Saliency Cards: A Framework to Characterize and Compare Saliency Methods

Angie Boggust*
aboggust@csail.mit.edu
MIT CSAIL
Cambridge, Massachusetts, USA

Harini Suresh*
hsuresh@mit.edu
MIT CSAIL
Cambridge, Massachusetts, USA

Hendrik Strobelt
IBM Research
Cambridge, Massachusetts, USA

John Guttag
MIT CSAIL
Cambridge, Massachusetts, USA

Arvind Satyanarayan
MIT CSAIL
Cambridge, Massachusetts, USA

ABSTRACT

Saliency methods are a common class of machine learning interpretability techniques that calculate how important each input feature is to a model's output. We find that, with the rapid pace of development, users struggle to stay informed of the strengths and limitations of new methods and, thus, choose methods for unprincipled reasons (e.g., popularity). Moreover, despite a corresponding rise in evaluation metrics, existing approaches assume universal desiderata for saliency methods (e.g., faithfulness) that do not account for diverse user needs. In response, we introduce *saliency cards*: structured documentation of how saliency methods operate and their performance across a battery of evaluative metrics. Through a review of 25 saliency method papers and 33 method evaluations, we identify 10 attributes that users should account for when choosing a method. We group these attributes into three categories that span the process of computing and interpreting saliency: *methodology*, or how the saliency is calculated; *sensitivity*, or the relationship between the saliency and the underlying model and data; and, *perceptibility*, or how an end user ultimately interprets the result. By collating this information, saliency cards allow users to more holistically assess and compare the implications of different methods. Through nine semi-structured interviews with users from various backgrounds, including researchers, radiologists, and computational biologists, we find that saliency cards provide a detailed vocabulary for discussing individual methods and allow for a more systematic selection of task-appropriate methods. Moreover, with saliency cards, we are able to analyze the research landscape in a more structured fashion to identify opportunities for new methods and evaluation metrics for unmet user needs.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Software and its engineering** → **Documentation**; **Software evolution**; • **General and reference** → **Evaluation**; • **Human-centered computing** → **User studies**.

KEYWORDS

saliency cards, transparency, interpretability, documentation, saliency

ACM Reference Format:
Angie Boggust, Harini Suresh, Hendrik Strobelt, John Guttag, and Arvind Satyanarayan. 2023. Saliency Cards: A Framework to Characterize and Compare Saliency Methods. In *2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT '23), June 12–15, 2023, Chicago, IL, USA.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3593013.3593997

1 INTRODUCTION

As machine learning (ML) systems are deployed in real-world contexts, stakeholder interviews [12, 76], design best practices [5], and legal frameworks [25] have underscored the need for explainability. Saliency methods — a class of explanation methods that identify input features important to an ML model's output — are frequently used to provide explanations. Saliency methods have helped ML researchers evaluate new models [13, 44], clinicians make AI-assisted patient care decisions [59], and users deploy fair and generalizable models [62]. As the popularity of saliency methods has grown, the number and diversity of saliency methods have correspondingly increased [24, 28, 39, 49, 57, 62, 66, 69–72]. However, since each saliency method operates differently according to its algorithmic goals, conflicts have arisen as multiple methods can produce varying explanations for the same model and input [43].

Researchers have proposed metrics to evaluate the effectiveness of saliency methods [1, 23, 47, 75, 82]. While promising, these approaches assume universal desiderata all saliency methods must achieve to be worth considering. Evaluations are often described as "tests" [23], "sanity checks" [1], or "axioms" [72], suggesting the existence of an *ideal* saliency method that passes every possible evaluation. However, this framing overlooks that saliency methods are *abstractions* of model behavior. They cannot offer a complete or wholly accurate reflection of a model's behavior (akin to a printout of model weights), so saliency methods must decide what information to preserve and sacrifice. Critically, saliency method abstraction decisions are motivated by downstream human-centric goals, such as generalizability [15, 49, 62], algorithmic simplicity [15], or perceptibility [39, 70]. Given the rich diversity in end-user expertise and needs [43, 74], it is unlikely that one set of abstraction decisions

*Both authors contributed equally to this research.

FAccT '23, June 12–15, 2023, Chicago, IL, USA

ACM ISBN 979-8-4007-0192-4/23/06.
https://doi.org/10.1145/3593013.3593997

will support all users, contexts, and tasks. How, then, should end users characterize and compare saliency methods to choose the most suitable one for their particular application?

The lack of standardized documentation for saliency methods and evaluative metrics makes it challenging to determine the benefits and limitations of particular methods and identify differences between them. Without resources equivalent to *model cards* [51] or *datasheets* [29], users are left to reference a potential sequence of research papers, including the original saliency method and all subsequent evaluations. Given the influx of saliency method research, this is a prohibitively time-consuming process and is especially out-of-reach for the broad class of users without a research background in machine learning, like clinicians, lawmakers, and engineers. Moreover, this piecemeal assembly of information considers each evaluative result in isolation, making it challenging to reason about whether desirable properties of a saliency method (e.g., input sensitivity [80] and minimality [15]) may be in tension with one another. As a result, and as we find through a series of interviews, users currently select saliency methods in unprincipled manners, such as choosing a method based on its popularity instead of a thorough understanding of its strengths and limitations.

In response, we introduce *saliency cards*: a structured documentation of how a saliency method is designed, operates, and performs across evaluative metrics. Reflecting the diversity of user needs, we identify ten attributes of saliency methods that users may wish to consider when choosing a particular approach. To facilitate comprehension, we group these attributes into three categories corresponding to different parts of the process of computing and interpreting saliency: *methodology*, or how the saliency is calculated; *sensitivity*, or relationships between the saliency and the model, input, or label; and *perceptibility*, or how a user perceives the output saliency. By collating this information, saliency cards help surface a method's strengths and weaknesses more holistically than individual paper results. Moreover, by offering a standard structure and visual design, saliency cards allow users to more easily compare methods, and more carefully reason about tradeoffs that might otherwise have been unapparent in the method design.

To evaluate the usefulness of saliency cards, we conduct a semi-structured interview study with nine saliency method users from diverse backgrounds, including saliency method developers, ML researchers, radiologists, computational biologists, and consultants. We find that saliency cards help users systematically select a saliency method that meets their needs. While previously, users chose saliency methods based on popularity or familiarity, with saliency cards, users prioritized attributes based on their task requirements. Using the visual format of saliency cards, users efficiently analyzed their prioritized attributes and weighed tradeoffs between saliency methods to uncover the method best suited to their task. Saliency card attributes also provided a shared vocabulary to discuss saliency methods, enabling users to precisely communicate their preferences, regardless of their prior experience with machine learning. Standardized documentation enabled side-by-side comparison, revealing ripe areas for future work, including saliency methods designed for a specific set of priorities, additional evaluation metrics for understudied attributes, and customized evaluations based on a user's data and models.

Saliency card templates and examples are available at: https://github.com/mitvis/saliency-cards.

2 RELATED WORK

Saliency methods (often referred to as feature attribution methods) are popular techniques for explaining a machine learning model's decision. Given an input, model, and target label, saliency methods compute a feature-wise importance score describing each feature's influence on the model's output for the target label. However, each saliency method computes these feature importances differently. Gradient-based methods, such as guided backpropagation [71] and Grad-CAM [66], compute importance using the model's gradients. Perturbation-based methods, such as SHAP [49] and RISE [57], measure importance by modifying input features and measuring the model's response. And path-based methods, such as integrated gradients [72] and XRAI [39], compute feature importances by comparing model outputs for the actual input to a meaningless input. While these granular categorizations [52] sort saliency methods based on algorithmic differences, they do not capture the complete set of considerations. Two gradient-based methods can operate differently, apply to separate tasks, and have distinct usage considerations. Saliency cards expand upon existing categorization criteria by documenting the saliency method's algorithm as well as other usage considerations, such as its hyperparameters and how to set them, dependence on model architectures, computational constraints, and sources of non-determinism.

As the popularity of saliency methods has grown, a related line of research has begun evaluating saliency methods' *faithfulness* — i.e., their ability to accurately represent the model's decision-making process. These evaluations are varied and include measuring the impact of adversarial perturbations [30], model randomization [1], dataset shifts [40], and input dropout [63] on the saliency output. However, its common for saliency methods to pass some faithfulness tests while failing others [75]. These discrepancies reflect the fact that faithfulness is too broad of a goal for evaluating saliency methods. As abstractions of model behavior, saliency methods necessarily preserve and sacrifice information in service of other human-centered goals such as simplicity or perceptibility. Depending on the use case, a user may accept a method that performs poorly on an evaluation that is low-priority for their task. Saliency cards lend a structure to these existing evaluating methods by splitting the concept of faithfulness into granular attributes that can inform tradeoffs and usage decisions for specific tasks. Saliency cards group evaluation methods that test similar concepts, such as a saliency method's response to label perturbations [1, 79], while drawing distinctions between those that test other factors, such as consistency across models [23] or saliency localization [83].

Saliency cards are complementary to documentation standards for machine learning datasets [8, 11, 22, 29, 33, 35, 50, 60] and models [2, 19, 50, 51, 65, 67]. These transparency artifact have been widely adopted by the ML community and led to increased trust and dissemination [26, 27, 50]. However, there is no standardized procedure for releasing saliency methods. Consequently, when selecting an appropriate saliency method, users must reference the original paper and subsequent evaluations to understand the algorithm, its advantages and limitations, and how to use it effectively.

This process is time-consuming for all users, but it is particularly prohibitive for users with little academic ML training, such as clinicians, lawmakers, and engineers. Saliency cards address this gap by providing a documentation structure and surfacing useful considerations about saliency methods to a range of stakeholders.

3 THE STRUCTURE OF SALIENCY CARDS

Saliency cards summarize information about a saliency method, including its developers, design goals, input/model/user assumptions, dependencies, usage considerations, benefits and limitations, and performance across various evaluations. The cards are structured as a series of attributes, grouped into three categories. We derived these attributes and categories through reviewing literature on saliency method algorithms, evaluation metrics used to assess them, and commonly-cited desiderata for model explanations. Our approach entailed iteratively 1) finding commonalities across methods, stated desiderata, and evaluative metrics, 2) distilling attributes that captured these commonalities, and 3) applying these attributes to compare and understand a broad set of saliency methods. This process yielded ten attributes grouped into three higher-level categories: *methodology*, or how the method operates; *sensitivity*, or how the method responds to changes in the model or data; and *perceptibility*, or how a human interprets the result of the method (Fig. 1). For each attribute, we provide a visual example (Fig. 3) and real-world application from our user study (Sec. 4).

3.1 Methodology

The methodology section of a saliency card summarizes the method's algorithm, including references to demos, papers, and/or implementations. In addition, this section includes information about five attributes related to how the saliency method operates: determinism, hyperparameter dependence, model agnosticism, computational efficiency, and semantic directness. By documenting these details, the methodology section provides an informative summary that helps users understand if a saliency method applies to their task.

3.1.1 Determinism. Determinism measures if a saliency method will always produce the same saliency map given a particular input, label, and model. Some saliency techniques, like LIME [62] and SHAP [49], are non-deterministic, so running them multiple times can produce significantly different results. Non-determinism can be introduced in the algorithm's definition; for instance, by computing the many random masks used by RISE [57]. It can also result from stochastic hyperparameters, like using random noise as a baseline value for integrated gradients [72].

Understanding how deterministic a saliency method is can impact if and how users apply the method. For example, in a clinical diagnosis task, a non-deterministic method could result in saliency maps highlighting slightly different portions of the radiograph. The radiologist we interviewed (U9) worried that these variations may have significant consequences given that small areas of the image can be integral to the diagnosis. Looking only at a single saliency map could skew a radiologist's judgment, while interpreting multiple maps together may be too time-consuming for the task. Thus, a user might choose to prioritize a deterministic saliency method in this setting. On the other hand, non-determinism can provide helpful additional context by surfacing multiple reasons for a model's decision. Just as humans can provide multiple correct justifications for their decisions, models likely have multiple feature sets sufficient to make correct and confident decisions. For example, in an image classification task, a model may correctly identify the object in the image using independent subsets of the object. A model developer we interviewed, U5, uses saliency to uncover spurious correlations between uninformative inputs and correct model outputs, so they were interested in leveraging non-determinism to surface all possible model justifications.

3.1.2 Hyperparameter Dependence. Hyperparameter dependence measures a saliency method's sensitivity to user-specified parameters. Some methods, like vanilla gradients [24, 69], do not have hyperparameter settings or require user intervention. However, other methods, like integrated gradients [72], require users to set consequential hyperparameters whose ideal values vary drastically depending on the task.

By documenting a method's hyperparameter dependence, saliency cards inform users of consequential parameters and how to set them appropriately. Using default parameter values can be misleading if users do not have sufficient resources or expertise to devote to hyperparameter tuning. Similarly, confusing results can arise if the hyperparameters were chosen based on a particular dataset but deployed in a setting with significant distribution shift. In situations like this, it makes sense for users to prioritize methods with low hyperparameter dependence. The radiologist (U9) prioritized hyperparameter dependence because they worried an incorrectly set parameter could have life-or-death consequences. Medical data often differs from the research datasets parameters are tuned on, so they worried a software vendor might not set consequential parameters appropriately. For example, integrated gradients [72] computes feature importance by interpolating between a "baseline" parameter and the actual input. A common practice is to use a baseline of all zeroes; however, using a zero baseline in x-ray images can be misleading and potentially harmful. In x-rays, black pixels convey meaning, such as indicating a bone fracture. If a software vendor uses the default zero baseline (black), integrated gradients will indicate that the fracture pixels are unimportant. The saliency card for integrated gradients would describe this hyperparameter dependence, alerting users to choose an appropriate baseline value or select a saliency method with less hyperparameter dependence. On the other hand, when researchers have dedicated appropriate time and resources to hyperparameter tuning, it could be preferable to use a method dependent on hyperparameters, like SmoothGrad [70], because it has other desired attributes, like minimality (Sec. 3.3.1).

3.1.3 Model Agnosticism. Model agnosticism measures how much access to the model a saliency method requires. Model agnostic methods, such as SHAP [49], treat the underlying model as a black box, relying only on its input and output. On the other hand, model-dependent methods, like Grad-CAM [66], require access to model internals. Model-dependent methods may have specific requirements, such as differentiability (e.g., gradient-based methods) or a specific model architecture (e.g., Grad-CAM requires a CNN [66]).

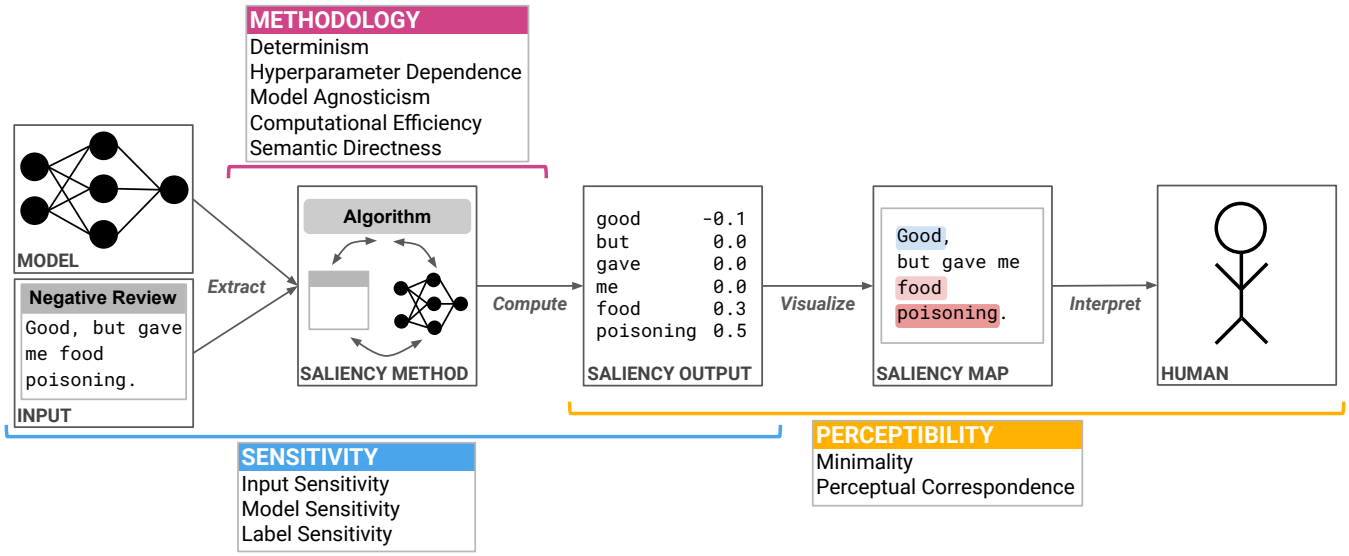


Figure 1: Saliency cards characterize saliency methods via ten user-centric attributes grouped into three categories corresponding to different phases of the interpretation process. *Methodology* attributes describe how the saliency is computed, *sensitivity* attributes express relationships between the saliency and its inputs, and *perceptibility* attributes measure human perception of the saliency.

By documenting a saliency method's model agnosticism, saliency cards help users identify whether a particular method might be incompatible with their use case. For instance, computational biologists, like U8, use proprietary machine learning models hosted through web-based GUIs, where users upload their input data and the model returns its predictions. In this setting, it is impossible to use a saliency method that requires access to model internals. Users, including U2, might also need a model agnostic saliency method for use cases that involve comparing saliency maps across different kinds of models. On the other hand, model agnosticism is not priority for users, like U1, who only consider specific model architectures. For these use cases, a user might choose a model-dependent method, like Grad-CAM [66], to prioritize other attributes essential to their task, like increased model sensitivity (Sec. 3.2.3).

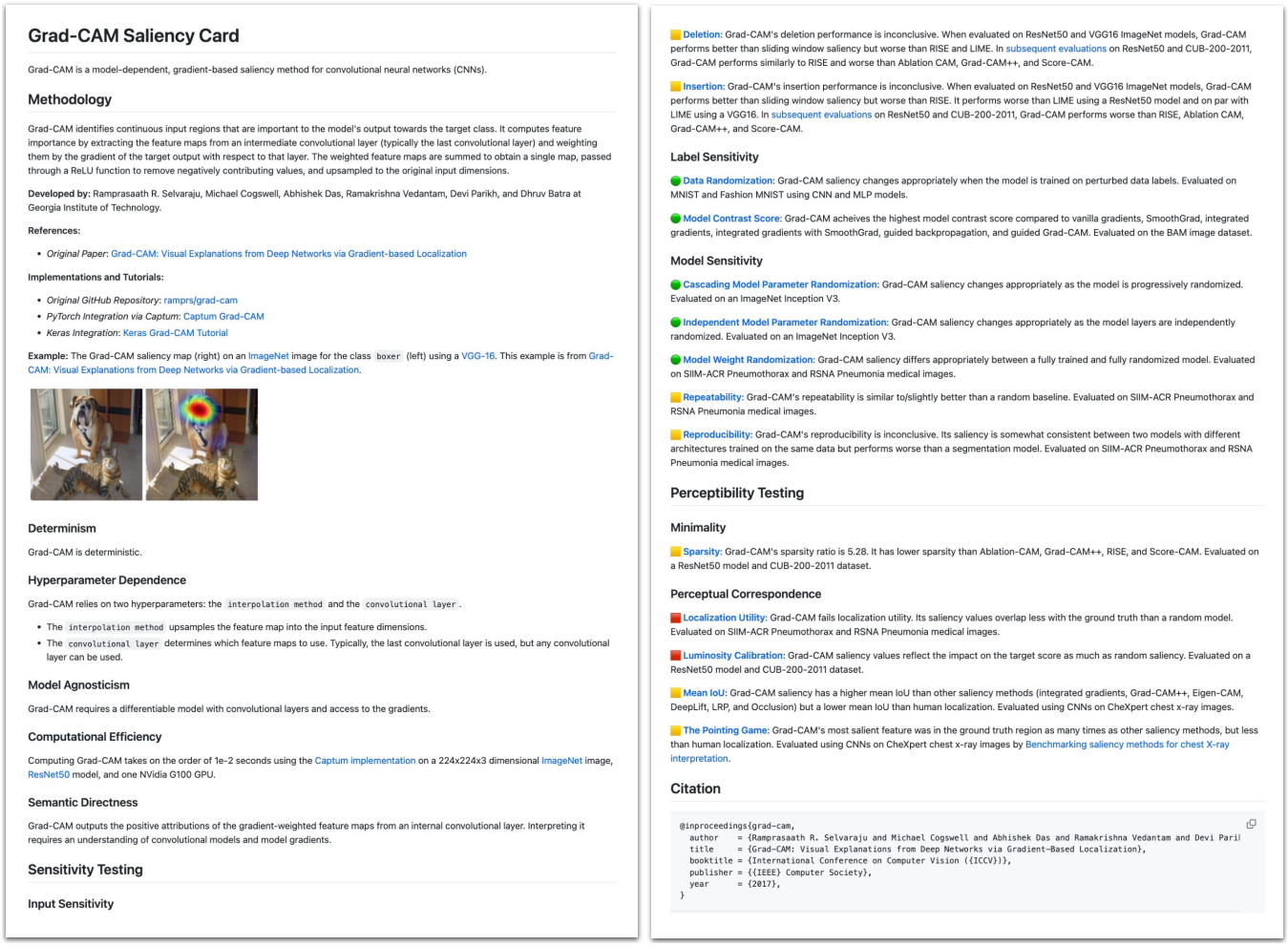
3.1.4 Computational Efficiency. Computational efficiency measures how computationally intensive it is to produce the saliency map. Different methods vary widely in their computational efficiency. For example, perturbation-based methods, like meaningful perturbations [28], are often more computationally intensive than simple gradient-based methods, like guided backpropagation [71].

By documenting computational efficiency, saliency cards help users determine whether running a particular saliency method is feasible in their setting. Several details about the task — including the number of saliency maps to compute, the number of models to compare, the amount of available computational resources, the size of the input data — all play a role in whether or not to prioritize computational efficiency. For example, some users, like Pfau et al. [58] and U1, use saliency maps to compute aggregate statistics about their model's behavior across an entire dataset. Given the size of existing machine learning datasets, this task could require computing hundreds of thousands of saliency maps to analyze a single model. In this setting, users may need to prioritize computational efficiency over other attributes.

3.1.5 Semantic Directness. Saliency methods abstract different aspects of model behavior, and semantic directness represents the complexity of this abstraction. For example, the saliency map computed by SIS represents a minimal set of pixels necessary for a confident and correct prediction [15]. Meanwhile, LIME's saliency map represents the learned coefficients of a local surrogate model trained to mimic the original model's decision boundary [62].

Semantically direct saliency methods do not require understanding complex algorithmic mechanisms such as surrogate models (e.g., as in LIME [62]) or accumulated gradients (e.g., as in integrated gradients [72]). As a result, their outputs might be more intuitive to users without formal ML expertise. Documenting semantic directness in a saliency card can help users prioritize it for tasks where the saliency maps will be interpreted by people with varying backgrounds. For example, in our interviews, users who worked in mixed-experience teams, like U8's biologist coworkers and U7's business clients, prioritized semantic directness to help them efficiently and effectively communicate their results. However, in cases where the interpreter can understand the saliency method or is comfortable not understanding the algorithm, semantic directness may not be a priority. We might choose a saliency method that is not semantically direct, like SHAP [49] (which defines feature importances as game theoretic Shapley values), because it improves other attributes, like perceptual correspondence (Sec. 3.3.2).

Semantic directness is a methodological attribute because it describes the complexity of the saliency method's algorithm. However, it is also related to perceptibility (Sec. 3.3) because it informs what types of users will use a saliency method and how they will interpret its results. We choose to delineate the methodology and



Grad-CAM Saliency Card

Grad-CAM is a model-dependent, gradient-based saliency method for convolutional neural networks (CNNs).

Methodology

Grad-CAM identifies continuous input regions that are important to the model's output towards the target class. It computes feature importance by extracting the feature maps from an intermediate convolutional layer (typically the last convolutional layer) and weighting them by the gradient of the target output with respect to that layer. The weighted feature maps are summed to obtain a single map, passed through a ReLU function to remove negatively contributing values, and upsampled to the original input dimensions.

Developed by: Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra at Georgia Institute of Technology.

References:

- *Original Paper:* Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

Implementations and Tutorials:

- *Original GitHub Repository:* ramprs/grad-cam
- *PyTorch Integration via Captum:* Captum Grad-CAM
- *Keras Integration:* Keras Grad-CAM Tutorial

Example: The Grad-CAM saliency map (right) on an ImageNet image for the class `boxer` (left) using a VGG-16. This example is from Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization.

Determinism

Grad-CAM is deterministic.

Hyperparameter Dependence

Grad-CAM relies on two hyperparameters: the `interpolation method` and the `convolutional layer`.

- The `interpolation method` upsamples the feature map into the input feature dimensions.
- The `convolutional layer` determines which feature maps to use. Typically, the last convolutional layer is used, but any convolutional layer can be used.

Model Agnosticism

Grad-CAM requires a differentiable model with convolutional layers and access to the gradients.

Computational Efficiency

Computing Grad-CAM takes on the order of 1e-2 seconds using the Captum implementation on a 224x224x3 dimensional ImageNet image, ResNet50 model, and one NVidia G100 GPU.

Semantic Directness

Grad-CAM outputs the positive attributions of the gradient-weighted feature maps from an internal convolutional layer. Interpreting it requires an understanding of convolutional models and model gradients.

Sensitivity Testing

Input Sensitivity

🟨 **Deletion:** Grad-CAM's deletion performance is inconclusive. When evaluated on ResNet50 and VGG16 ImageNet models, Grad-CAM performs better than sliding window saliency but worse than RISE and LIME. In subsequent evaluations on ResNet50 and CUB-200-2011, Grad-CAM performs similarly to RISE and worse than Ablation CAM, Grad-CAM++, and Score-CAM.

🟨 **Insertion:** Grad-CAM's insertion performance is inconclusive. When evaluated on ResNet50 and VGG16 ImageNet models, Grad-CAM performs better than sliding window saliency but worse than RISE. It performs worse than LIME using a ResNet50 model and on par with LIME using a VGG16. In subsequent evaluations on ResNet50 and CUB-200-2011, Grad-CAM performs worse than RISE, Ablation CAM, Grad-CAM++, and Score-CAM.

Label Sensitivity

🟩 **Data Randomization:** Grad-CAM saliency changes appropriately when the model is trained on perturbed data labels. Evaluated on MNIST and Fashion MNIST using CNN and MLP models.

🟩 **Model Contrast Score:** Grad-CAM acheives the highest model contrast score compared to vanilla gradients, SmoothGrad, integrated gradients, integrated gradients with SmoothGrad, guided backpropagation, and guided Grad-CAM. Evaluated on the BAM image dataset.

Model Sensitivity

🟩 **Cascading Model Parameter Randomization:** Grad-CAM saliency changes appropriately as the model is progressively randomized. Evaluated on an ImageNet Inception V3.

🟩 **Independent Model Parameter Randomization:** Grad-CAM saliency changes appropriately as the model layers are independently randomized. Evaluated on an ImageNet Inception V3.

🟩 **Model Weight Randomization:** Grad-CAM saliency differs appropriately between a fully trained and fully randomized model. Evaluated on SIIM-ACR Pneumothorax and RSNA Pneumonia medical images.

🟨 **Repeatability:** Grad-CAM's repeatability is similar to/slightly better than a random baseline. Evaluated on SIIM-ACR Pneumothorax and RSNA Pneumonia medical images.

🟨 **Reproducibility:** Grad-CAM's reproducibility is inconclusive. Its saliency is somewhat consistent between two models with different architectures trained on the same data but performs worse than a segmentation model. Evaluated on SIIM-ACR Pneumothorax and RSNA Pneumonia medical images.

Perceptibility Testing

Minimality

🟨 **Sparsity:** Grad-CAM's sparsity ratio is 5.28. It has lower sparsity than Ablation-CAM, Grad-CAM++, RISE, and Score-CAM. Evaluated on a ResNet50 model and CUB-200-2011 dataset.

Perceptual Correspondence

🟥 **Localization Utility:** Grad-CAM fails localization utility. Its saliency values overlap less with the ground truth than a random model. Evaluated on SIIM-ACR Pneumothorax and RSNA Pneumonia medical images.

🟥 **Luminosity Calibration:** Grad-CAM saliency values reflect the impact on the target score as much as random saliency. Evaluated on a ResNet50 model and CUB-200-2011 dataset.

🟨 **Mean IoU:** Grad-CAM saliency has a higher mean IoU than other saliency methods (integrated gradients, Grad-CAM++, Eigen-CAM, DeepLift, LRP, and Occlusion) but a lower mean IoU than human localization. Evaluated using CNNs on CheXpert chest x-ray images.

🟨 **The Pointing Game:** Grad-CAM's most salient feature was in the ground truth region as many times as other saliency methods, but less than human localization. Evaluated using CNNs on CheXpert chest x-ray images by Benchmarking saliency methods for chest X-ray interpretation.

Citation

```
@inproceedings{grad-cam,
  author    = {Ramprasaath R. Selvaraju and Michael Cogswell and Abhishek Das and Ramakrishna Vedantam and Devi Pari
  title     = {Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization},
  booktitle = {International Conference on Computer Vision ({ICCV})},
  publisher = {{IEEE} Computer Society},
  year      = {2017},
}
```

Figure 2: Saliency cards begin with a summary of the saliency method's algorithm, an example output, and references. Next, the card describes the saliency method's methodology, including its determinism, hyperparameter dependence, model agnosticism, computational efficiency, and semantic directness. Finally, the saliency card summarizes the method's performance on sensitivity (input, label, and model sensitivity) and perceptibility (minimality and perceptual correspondence) tests.

perceptibility attributes based on their testability — i.e., which attributes have been quantified in the literature. Currently, semantic directness is purely descriptive, and no saliency evaluations test for it. As evaluations are developed (Sec. 4.3), we may discover that semantic directness is testable and part of perceptibility or confirm it is part of methodology because it is unique to each user.

3.2 Sensitivity Testing

The sensitivity section of a saliency card details whether a saliency method changes in proportion to meaningful changes in the model and data. While prior work has treated sensitivity as an overarching goal of all saliency methods, saliency cards break sensitivity down into three independent attributes: input sensitivity, label sensitivity, and model sensitivity. Saliency methods often perform differently across each of these attributes. By disentangling them, saliency cards let users prioritize and make trade offs based on their needs.

Unlike the methodology section, which contains descriptive information about each attribute, the sensitivity section enumerates experimental evaluations for each attribute, documenting the results and linking to the original evaluation. As a result, saliency cards provide a glanceable representation of complex evaluations, helping users of all backgrounds evaluate a method's sensitivity.

3.2.1 Input Sensitivity. Input sensitivity measures if a saliency method accurately reflects the model's sensitivity to transformations in the input space. If an input transformation meaningfully affects the model's output (e.g., an adversarial attack), the saliency method should assign importance to the transformed features. Otherwise, if an input transformation does not change the model's output (e.g., a noise-based perturbation), then the saliency method should not assign additional importance to the modified features.

Documenting a saliency method's input sensitivity is essential for tasks that use saliency to understand the impact of input

changes, such as studying a model's robustness to adversarial attacks [37], or its reliance on sensitive input features (e.g., race or age) [10]. For instance, U3, a fairness researcher, tests models for *counterfactual fairness* [45] by confirming the model's decision on an input is the same even if its sensitive attributes are inverted. Without input sensitivity, a saliency method might indicate that a sensitive feature is unimportant even if the model's decision is counterfactually unfair. As a result, the method could mislead a user to trust and deploy a discriminatory model. However, in other cases, users might choose to trade off input sensitivity for higher priority attributes. For example, we interviewed ML researchers (U5, U6) who only work in controlled settings that are less dependent on input changes, including analysis of in-distribution data or comparison of different models on the same data.

Saliency cards contain the results of numerous existing input sensitivity tests [3, 4, 6, 15, 17, 23, 31, 34, 38, 40, 57, 63, 72, 80]. Several of these tests measure whether the model's output changes significantly in response to perturbations of salient features and little in response to changes in non-salient features. Extensions of them evaluate if the salient input features are sufficient to train a new, equally performant model [15, 34]. Other metrics test input sensitivity by measuring the saliency map's response to uninformative transformations of the input dataset [40] or adding small amounts of noise to the inputs [3, 80]. Saliency cards should report the results of various input sensitivity tests to communicate a comprehesive overview of a saliency method's input sensitivity.

3.2.2 Label Sensitivity. Saliency maps are computed with respect to a particular target label. Label sensitivity measures the saliency method's response to changes to that target label.

Documenting label sensitivity is important for tasks that evaluate model behavior by changing the target label. For example, in computational immunology, users like U8 train multi-task models that take in an antibody and predict how well it binds to several different target molecules [48]. In this setting, U8 wants to compare which antibody features are important to the model's prediction and how they differ across multiple targets. To do so, they compare saliency maps on the same antibody with respect to different target labels. Without label sensitivity, the saliency maps may inaccurately reflect the difference in model reasoning for different target molecules. Since antibody binding is not fully understood, label insensitive saliency methods could mislead users about which features are biologically important.

Saliency cards include the results of label sensitivity tests. Existing label sensitivity evaluations swap labels in controlled ways and measure changes in the saliency map. Tests include measuring how the saliency maps change in response to label randomization [1] and when switching from fine-grained to coarse-grained labels [79].

3.2.3 Model Sensitivity. Model sensitivity measures if the output of a saliency method is sensitive to meaningful changes to the model parameters. If we modify the model significantly (e.g., by randomizing its weights), then the output of a model sensitive saliency method will reflect that change.

Model sensitivity is crucial for tasks that compare models. For example, we interviewed an ML researcher (U2) who evaluates their training procedure by comparing saliency maps across models from different epochs. They prioritized model sensitive methods to accurately reflect meaningful model changes. However, users may trade off model sensitivity for other desirable attributes in tasks where the model is not changing. For instance, if a user is focused on the impact of modifying input features, they might trade off model sensitivity to get a highly input sensitive method.

Saliency cards document a method's model sensitivity using existing model sensitivity tests. Testing for model sensitivity involves evaluating how the output of a saliency method changes in response to known similarities or differences between models. Some methods test the consistency of saliency maps between similar models [9, 23, 72], while others confirm that saliency maps sufficiently change due to model randomization [1] or combination [72].

3.3 Perceptibility Testing

The perceptibility section of a saliency card describes attributes of a saliency method related to human perception of the saliency map. Perceptibility is split into two attributes: minimality and perceptual correspondence. Minimality captures the idea that a saliency map should highlight a minimal set of features, which can be an important consideration for users who visually analyze saliency maps. Perceptual correspondence measures if the interpreted signal in the saliency map reflects the features' importance and does not introduce misleading visual artifacts. As with the sensitivity attributes, saliency cards summarize the results of perceptibility tests.

3.3.1 Minimality. Minimality measures how many unnecessary features are given a significant value in the saliency map. Methods, such as vanilla gradients [69], that attribute importance to many input features can produce noisy results that are difficult to interpret. On the other hand, methods like XRAI [39] incorporate minimality into their algorithms by attributing importance to higher-level feature regions instead of many individual features.

Documenting a method's minimality can alert users to the amount of noise they can expect in the saliency map. For example, minimality is particularly important when interpreting complex high-dimensional data, such as the long amino acid sequences our computational biologist interviewee (U8) uses. In this case, it might be prohibitively difficult or time-consuming to interpret hundreds of important features, and a noisy saliency map could risk obscuring underlying signal. However, some users may prefer a less minimal saliency method depending on their task. For instance, some bioinformatics tasks operate on input sequences containing only nine amino acids [7]. In this setting, users might actually prefer a less minimal method that reveals every amino acid that influences the model's prediction, since it provides a complete picture of important features and is relatively easy to analyze.

Saliency cards report a saliency method's performance on minimality metrics. Some minimality metrics compare the maximum and mean saliency values [31]. The higher the ratio, the more minimal the saliency map is, since it is focused on only a few input features. Minimality can also be evaluated by testing if any salient feature can be removed without the model's confidence dropping below a chosen threshold [15]. For methods that are not inherently minimal, applying SmoothGrad [70] can increase minimality by reducing the noise present in the saliency map; however, it may also impact other attributes of the original method.

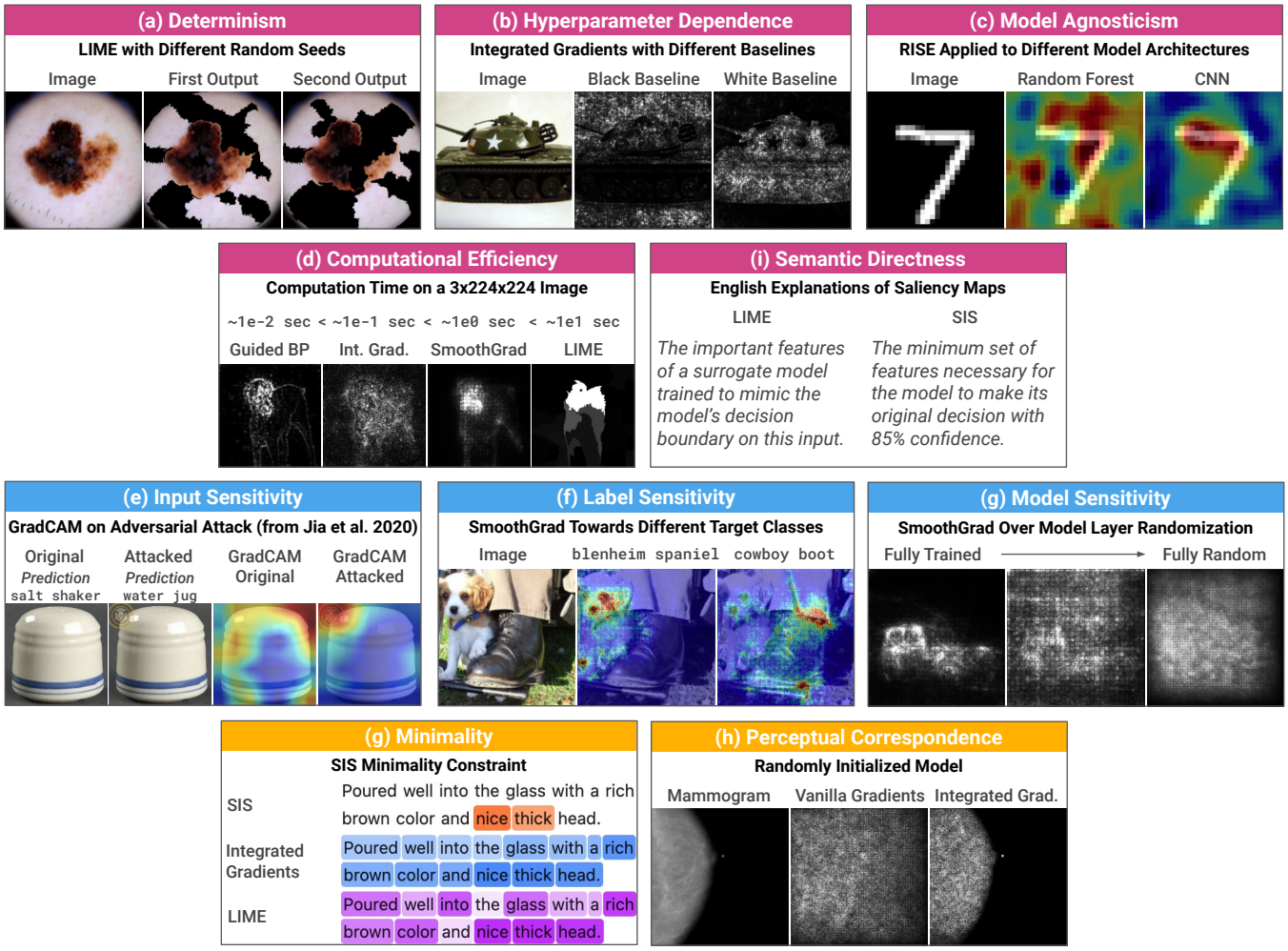


Figure 3: The ten saliency card attributes represent saliency method characteristics that can help users choose a task-appropriate saliency method. Each attribute is shown above alongside a descriptive example that communicates the principle.

3.3.2 Perceptual Correspondence. Perceptual correspondence measures if the perceived signal in the saliency map accurately reflects the feature importance. For example, saliency maps with high perceptual correspondence should not contain visual artifacts that lead users to incorrectly infer signal.

Saliency cards document perceptual correspondence because it is crucial for high-risk settings where a misleading signal could provide an unwarranted justification for decisions or lead users down incorrect paths. However, perceptual correspondence may be less critical if the saliency method is used for large-scale analyses of model behavior that will aggregate one-off artifacts (i.e., when such artifacts occur arbitrarily and are washed out by averaging). For instance, users, like U1, who aggregate metrics on saliency maps across an entire dataset may be willing to prioritize computational efficiency over perceptual correspondence.

Saliency cards report the results of perceptual correspondence tests. The most direct perceptual correspondence metric measures if a feature's visualized luminosity is calibrated to its importance [31]. Other perceptual correspondence evaluations compare salient features to human-defined ground truth features [9, 23, 83]. However, these metrics should only be used to evaluate saliency methods in test settings where the model is known to rely on ground truth features. Even high-performing models have been shown to rely on spurious features [14, 78]. In those cases, a perceptually correspondent saliency method would correctly highlight features outside of the ground truth but get low perceptual correspondence scores.

4 EVALUATIVE USER STUDIES

Through nine semi-structured user interviews, we evaluate saliency cards to understand how they can help users understand, compare, and select methods appropriate for their tasks. We recruited participants from within our professional network (4/9) and through an open call in our organizations and on Twitter (5/9). Participants came from diverse backgrounds (including research, radiology, computational biology, and consultancy) and had varying levels of machine learning expertise and familiarity with saliency methods.

Fig. 4 illustrates user demographics, describes their saliency method use cases, and summarizes the results of our interview studies.

Eight participants (U1–U8) had used saliency methods in some capacity. With these participants, we began by asking open-ended questions about their experience with saliency methods, such as "*What tasks do you use saliency methods for?*", "*How do you decide which saliency method to use?*", and "*What do you do if saliency methods disagree?*". Next, we discussed each saliency method attribute. We asked participants to describe if and how each attribute was important to their task and rank the attributes by importance. Finally, we walked users through example saliency cards (Fig. 2) and had users give us feedback on the design and usefulness.

The radiologist (U9) did not have experience using saliency methods but was interested in their application to medical decision-making. Since they were less familiar with ML, we structured this conversation slightly differently. We discussed five attributes — determinism, hyperparameter dependence, semantic directness, minimality, and perceptual correspondence. For each, we provided a definition, showed radiology examples demonstrating its implications (e.g., saliency with and without minimality), and discussed if and when it would be important to them in a clinical setting.

We conducted 30–60min interviews via video chat and compensated users with $30 Amazon gift cards. Our study received an IRB exemption from our organization. We obtained informed consent from participants, stored user data securely, and anonymized user details in the paper. See Sec. A.3 for additional study details.

4.1 Saliency Cards Help Users Select Task-Appropriate Saliency Methods

Despite experience with a broad range of saliency methods, including LIME [62] (U3–U7), vanilla gradients [24, 69] (U5, U6, U8), integrated gradients [72] (U4–U6), Grad-CAM [66] (U1, U2), SHAP [49] (U3–U5), and SmoothGrad [70] (U5), users chose saliency methods based on their popularity in prior work (U1–U8) and ease of implementation (U2, U7). Users rarely considered algorithmic differences or evaluated the suitability of a particular method for their task. As a result, users were often unsure if their chosen saliency method was indeed appropriate for their task and worried that a different method could produce more accurate results. Consequently, users wanted a more principled selection strategy based on formal evaluations but found extracting insight from existing documentation tedious. As U6 described, given that "*new methods come out every day*" and "*reading all the papers is a difficult task that takes a lot of time,*" even researchers find it challenging to acquire the knowledge needed to select saliency methods well-suited to their tasks.

In contrast, we found that the attribute-based structure of saliency cards allowed users to more systematically select saliency methods based on properties important to their task. Users prioritized each attribute based on their task requirements, experiences and preferences, and the expectations of their teammates. For example, U1 prioritized computational efficiency because their research requires them to compute saliency maps for every input in their dataset. An inefficient saliency method would be incompatible with their model evaluation process and prevent them from quickly iterating on model design choices. U9 prioritized determinism based on their personal experience. They were uncomfortable interpreting non-deterministic saliency maps because they do not encounter non-determinism in other medical technologies. When prioritizing attributes, users in applied domains also needed to consider their teammates' expertise. For instance, U7 prioritized semantic directness to help them communicate results to business clients without ML experience. After prioritizing attributes, users utilized the visual saliency card documentation to juxtapose attribute summaries and pick a well-suited saliency method in just a few minutes.

Our user study also revealed that user priorities often differ or conflict — a surprising finding given that existing evaluations are often framed as "tests" every saliency method should pass [1, 23, 72]. While one user would prioritize an attribute, another would deprioritize or explicitly not desire that attribute. For example, U8 prioritized minimality because they train machine learning models on long amino acid sequences, and their biochemist coworkers interpret the saliency method results. Without minimality, confirming the models have learned biologically meaningful features could require the biochemists to analyze the interactions between potentially hundreds of amino acids. However, U4 explicitly preferred a less minimal method. They worried that a minimal saliency method might only highlight the features necessary for the model's prediction. Since they use saliency methods to manually analyze a few inputs, they want to view every feature relevant to the model's prediction to ensure their models do not learn spurious correlations. Even users in similar roles had different priorities. For instance, despite both being researchers who use saliency methods to analyze model behavior, U2 and U6 viewed the importance of input sensitivity differently. U2 regularly tests models by perturbing background features, so without input sensitivity, a saliency method could incorrectly assign importance to changes the model considers unimportant. On the other hand, U6 did not care about input sensitivity because they only use in-distribution data and do not worry about noise or perturbations impacting the inputs. The frequency of conflicting priorities suggests there is not an ideal saliency method for every user and task. Thus, documentation is crucial to help users find a saliency method appropriate for their use case.

4.2 Saliency Card Attributes Provide a Detailed Vocabulary for Discussing Saliency Methods

Saliency cards provide a more precise attribute-based vocabulary that helps users communicate about saliency methods. At the start of our interviews, participants often cited *faithfulness* as an ideal attribute of saliency methods. Faithfulness broadly refers to a saliency method's ability to reflect model reasoning accurately and correlates with the saliency card's sensitivity attributes. However, after discussing the ten saliency card attributes, users had a more detailed language to describe saliency method characteristics. For example, U5 initially expected all saliency methods to achieve faithfulness. However, after working with saliency cards, U5 more precisely articulated that they expected saliency methods to be label and model sensitive. They did not care about a method's input sensitivity, even though it is typically considered part of faithfulness. As a saliency method developer, U5 needs to be able to communicate their exact design goals so users can understand the benefits, limitations, and appropriate use cases of the saliency method. If they described their saliency method as faithful, users could incorrectly assume it is

	U1	U2	U3	U4	U5	U6	U7	U8	U9
Role:	**ML Researcher**	**ML Researcher**	**ML Researcher**	**Saliency Developer**	**Saliency Developer**	**Saliency Developer**	**Consultant**	**Comp. Biologist**	**Radiologist**
Task:	Uses saliency methods to compute aggregate statistics about their model's behavior across entire datasets. Works on a machine learning team that only develops CNN models.	Uses saliency methods to measure the importance of each modality in multimodal models. Visually analyzes saliency during training. Uses many model architectures.	Applies saliency methods to study the fairness of machine learning models. Primarily uses small tabular datasets. Works in deployment settings with doctors and clinicians.	Designs saliency methods the help users visually analyze a few saliency maps to understand model behavior. Values method generalizability to different tasks.	Evaluates saliency methods. Studies how users respond to saliency methods with different properties. Expects users to analyze a few saliency maps on inputs from within their data distribution.	Designs saliency methods the help users visually analyze a few saliency maps to understand model behavior. Focused on 2D inputs, including images and audio spectrograms.	Incorporates saliency methods into user interfaces. Expected to provide quick solutions to clients. Works with clients who are not familiar with machine learning.	Develops saliency methods to interpret biological models. Uses large inputs and various model architectures. Biological colleagues interpret the saliency. Access to extensive compute resources.	Unfamiliar with saliency methods. Hopes to use ML model in clinical settings. Expects saliency to suggest parts of a diagnostic image to analyze. Used to dealing with noise and complexity.
Essential Attributes	Determinism Comp. Efficiency Input Sensitivity Label Sensitivity Model Sensitivity	Determinism Model Agnosticism Input Sensitivity Model Sensitivity	Model Agnosticism Input Sensitivity Label Sensitivity Model Sensitivity Percept. Correspond.	Input Sensitivity Label Sensitivity Model Sensitivity	Label Sensitivity Model Sensitivity Percept. Correspond.	Hyperparam. Depend. Label Sensitivity Model Sensitivity Minimality	Hyperparam. Depend. Comp. Efficiency Semantic Directness Input Sensitivity	Model Agnosticism Semantic Directness Minimality	Determinism Hyperparam. Depend. Percept. Correspond.
Nice-to-have Attributes	Hyperparam. Depend.	Hyperparam. Depend. Label Sensitivity	Determinism Hyperparam. Depend. Semantic Directness Minimality	Determinism Hyperparam. Depend. Model Agnosticism Semantic Directness Percept. Correspond.	Determinism Hyperparam. Depend. Semantic Directness Minimality	Determinism Model Agnosticism Comp. Efficiency	Model Agnosticism Label Sensitivity Model Sensitivity Minimality Percept. Correspond.	Determinism Hyperparam. Depend. Input Sensitivity Label Sensitivity Percept. Correspond.	
Unimportant Attributes	Model Agnosticism Semantic Directness Minimality Percept. Correspond.	Comp. Efficiency Semantic Directness Minimality Percept. Correspond.	Comp. Efficiency	Comp. Efficiency Minimality	Model Agnosticism Comp. Efficiency Input Sensitivity	Semantic Directness Input Sensitivity Percept. Correspond.	Determinism	Comp. Efficiency Model Sensitivity	Semantic Directness Minimality

Figure 4: We evaluate saliency cards through semi-structured interviews with researchers, scientists, and domain experts. Each user prioritized saliency method attributes differently based on the needs of their tasks. Saliency cards helped users select task-appropriate methods, communicate about saliency methods, and hypothesize new areas for future work.

input sensitive, deploy it in an inappropriate setting, and misinterpret the results. Using a shared attribute-based vocabulary, users and developers can better communicate about a saliency method's specific attributes, evaluative results, and prescribed use cases.

The saliency card attributes also helped lay users discuss saliency methods. Before our user study, U9 (a radiologist) had little experience with machine learning and was entirely unfamiliar with saliency methods. However, by using the vocabulary of saliency card attributes, our conversation revealed differences in their expressed needs and expectations in the literature about what lay users want in a saliency method. For example, minimality is often considered an essential attribute because it makes the visual saliency map easier to interpret [39, 70, 73]. However, U9 did not expect a saliency method to be minimal because they were accustomed to using noise in medical imaging to attenuate measurement uncertainty. Using the saliency card attributes gave U9 terminology they could use to communicate with ML experts and software vendors in charge of developing and deploying saliency methods. Without this language with which to communicate, radiologists might not as deeply engage in the deployment process, leaving ML experts to rely on incorrect assumptions about radiologists' expectations. However, with direct channels of communication, ML experts could work with radiologists to increase transparency in the deployment process, ensure they interpret saliency method results appropriately, and, even, develop new saliency methods explicitly designed for clinical imaging settings.

4.3 Saliency Cards Inspire Areas for Future Work and New Documentation Practices

The attribute summaries led users to ask new questions about evaluating saliency methods and to hypothesize future research directions. By documenting evaluation results for a saliency method, saliency cards reveal that particular attributes and methods have been more heavily evaluated than others. For instance, comparing the saliency cards for integrated gradients [72] (Fig. A2) and Grad-CAM [66] (Fig. 2) reveals that integrated gradients has been more rigorously tested for input sensitivity. Whereas previously, users would have had to extract evaluative results from multiple academic papers, saliency cards surface these discrepancies directly, inspiring users to hypothesize about Grad-CAM's performance on missing evaluations and express interest in completing the testing suite. Further, by categorizing individual evaluations, saliency cards expose that evaluations for the same attribute have varying testing strategies, such as testing meaningful [4] vs. noisy perturbations [40] or focusing on images [31] vs. natural language modalities [23]. Users were surprised to see the evaluation diversity, leading them to hypothesize new evaluation measures. For instance, some users were intrigued to run perceptibility tests on their data and models. As U5 put it, "*If I have a specific use case in mind, I want to see the metrics on that specific use case.*" They brainstormed ideas about integrating saliency cards into a suite of evaluations that generate customized saliency cards based on the user's model and datasets.

Inspecting some attributes revealed limitations of saliency cards and existing evaluations. Saliency cards group evaluations into user-centric attributes, but some attributes are challenging to test accurately. During our user study, U8 was skeptical that existing evaluations appropriately assessed model sensitivity. Model sensitivity evaluations test that a saliency method responds to meaningful model changes, but U8 argued that it is almost impossible to guarantee that a change to a black-box model is meaningful. For instance, a standard model sensitivity test measures the saliency method's response to layer randomization, but layer randomization might not be meaningful if that layer is redundant. In that case, layer randomization tests could incorrectly punish a model for not responding to an insignificant change. This issue might be solved

as additional research invents new evaluations, including model sensitivity tests. However, it could also be that some attributes, like model sensitivity, are too broad. Perhaps breaking model sensitivity down into more precise categorizations, like layer randomization sensitivity, would provide more straightforward documentation. Similarly, we expect the methodology attributes to evolve from open-ended descriptions to more consistent reports. For instance, the vocabulary used to describe computational efficiency may vary across saliency developers and research areas based on typical computing resources and dataset sizes. As more saliency methods are documented and more evaluations are developed, we expect the saliency card attributes and their descriptions will evolve to better characterize saliency methods, facilitate cross-card comparison, and communicate with users.

Saliency method developers were inspired to document their methods with saliency cards and hoped consistent and thorough documentation would increase method adoption. Good documentation can make saliency methods easier to understand and use, "*If you want people to use your method, your need to have them understand it.*" [U8]. Currently, saliency method developers have to generate documentation content that ranges from novel algorithmic decisions and implications in the paper to implementation considerations in the public code repository. This process can feel unprincipled, so developers were excited to have a template that fully captured critical considerations. For example, when developing their saliency method, U8 documented their method's computational efficiency and hyperparameter dependence in their code repository, explaining "*We tried to make our documentation accessible to users. I tried to do some of this, but in an ad hoc way, and I didn't hit all of these [attributes].*" They looked forward to adding additional documentation and making a saliency card for their method.

5 DISCUSSION AND LIMITATIONS

We present saliency cards, transparency documentation to describe, communicate, and compare saliency methods. While documentation in other parts of the machine learning pipeline has led to increased trust and appropriate use [26, 29, 50, 51], saliency methods do not have documentation standards. As a result, users we interviewed struggled to stay informed with the ever-increasing number of saliency methods, forcing them to choose saliency methods based on popularity instead of a thorough understanding of their benefits and limitations. In response, saliency cards characterize saliency methods based on ten user-centric attributes that describe important usage considerations. The saliency card attributes span different phases of the interpretation workflow, such as the saliency method's algorithmic properties, relationship to the model and data, and perceptibility by an end-user. We evaluate saliency cards in a user study with nine participants, ranging from radiologists with limited knowledge of machine learning to saliency method developers. With saliency cards, users prioritized attributes based on their task requirements, personal experience, and the expectations of their teammates, allowing them to select a saliency method appropriate for their needs and properly interpret its results. Further, the saliency card attributes provided users with a shared vocabulary to describe their needs and communicate about saliency methods without requiring extensive machine learning expertise.

Building saliency cards allowed us to analyze the research landscape, revealing areas for future work, such as task-specific saliency methods and evaluation metrics for under-evaluated attributes. By documenting and comparing the methodological attributes of various saliency methods (Table A1), we identify the potential for new saliency methods that meet specific user priorities and future studies on the latent relationships between attributes. Current saliency methods cannot achieve specific combinations of attributes. For example, none of the saliency methods we surveyed were model agnostic and computationally efficient because model agnosticism is commonly achieved through expensive repeated perturbations. Model agnosticism and computational efficiency were priorities for U6 and U7, but currently, they must sacrifice one when choosing a saliency method. New research could explore this gap, and others, by designing novel saliency methods that attain model agnosticism without forfeiting computational efficiency or proving that they are inexplicably inversely correlated.

Saliency cards also revealed gaps in evaluation research, including under-evaluated attributes and saliency methods (Table A2). For example, by compiling evaluative metrics for each attribute, we uncovered that there is far less research into how to measure perceptual correspondence, relative to other attributes such as input or model sensitivity. By identifying this gap, saliency cards prompt further research into how we might measure perceptual correspondence. Better understanding how people perceive saliency maps could then motivate the design of new saliency visualizations — e.g., that expand static heatmaps by dynamically overlaying multiple attributions [55] to explicitly communicate limitations and preemptively avoid implying unwarranted signal. Table A2 also reveals that some saliency methods (e.g., SIS [15]) have been evaluated less than others (e.g., integrated gradients [72]). While the sensitivity and perceptibility attributes report results from existing evaluations, evaluation papers typically only test a subset of existing saliency methods. As a result, our users found it challenging to compare saliency methods evaluated on different tests. Future work could run missing evaluations or design test suites that report a saliency method's results on existing tests.

We intend saliency cards to be *living artifacts* that start a conversation around saliency method documentation. To facilitate living documentation, we provide a public repository[1] containing saliency card templates, summaries of evaluations, and saliency cards for existing methods. The repository serves as a centralized location for users to reference saliency methods. As new saliency methods are developed to fulfill specific user needs, new saliency cards can be added to the repository. Existing saliency cards can be continually updated with additional evaluative results stemming from new evaluation metrics and the application of existing metrics to unevaluated saliency methods. As the saliency card repertoire expands, saliency card documentation will simultaneously evolve to support additional user needs. As signaled in our user studies, new evaluations may reveal that some saliency card attributes are too broad and need to be decomposed into constituent attributes that more precisely articulate the evaluative takeaways. Likewise, new attributes or categories may emerge as more users from various backgrounds begin to use saliency methods and communicate

[1] https://github.com/mitvis/saliency-cards

their task-specific priorities. By documenting saliency methods, we hope saliency cards support the continued rapid growth of saliency method research and evolve as needed alongside new developments.

ACKNOWLEDGMENTS

This work is supported by NSF Award #1900991, and via a grant from the MIT-IBM Watson AI Lab. H.S. was supported by the Kerr Fellowship. Research was also sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity Checks for Saliency Maps. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*. 9525–9536.

[2] David Adkins, Bilal Alsallakh, Adeel Cheema, Narine Kokhlikyan, Emily McReynolds, Pushkar Mishra, Chavez Procope, Jeremy Sawruk, Erin Wang, and Polina Zvyagina. 2022. Prescriptive and Descriptive Approaches to Machine-Learning Transparency. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–9.

[3] David Alvarez-Melis and Tommi S. Jaakkola. 2018. On the Robustness of Interpretability Methods. *CoRR* abs/1806.08049 (2018). http://arxiv.org/abs/1806.08049

[4] David Alvarez-Melis and Tommi S. Jaakkola. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 7786–7795.

[5] Saleema Amershi, Daniel S. Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi T. Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. ACM, 3.

[6] Marco Ancona, Enea Ceolini, A. Cengiz Öztireli, and Markus H. Gross. 2017. A unified view of gradient-based attribution methods for Deep Neural Networks. *CoRR* abs/1711.06104 (2017). arXiv:1711.06104 http://arxiv.org/abs/1711.06104

[7] Massimo Andreatta and Morten Nielsen. 2016. Gapped sequence alignment using artificial neural networks: application to the MHC class I system. *Bioinformatics* 32, 4 (2016), 511–517.

[8] Matthew Arnold, Rachel KE Bellamy, Michael Hind, Stephanie Houde, Sameep Mehta, Aleksandra Mojsilović, Ravi Nair, K Natesan Ramamurthy, Alexandra Olteanu, David Piorkowski, et al. 2019. FactSheets: Increasing trust in AI services through supplier's declarations of conformity. *IBM Journal of Research and Development* 63, 4/5 (2019), 6–1.

[9] Nishanth Thumbavanam Arun, Nathan Gaw, Praveer Singh, Ken Chang, Mehak Aggarwal, Bryan Chen, Katharina Hoebel, Sharut Gupta, Jay B. Patel, Mishka Gidwani, Julius Adebayo, Matthew D. Li, and Jayashree Kalpathy-Cramer. 2020. Assessing the (Un)Trustworthiness of Saliency Maps for Localizing Abnormalities in Medical Imaging. *CoRR* abs/2008.02766 (2020).

[10] Aparna Balagopalan, Haoran Zhang, Kimia Hamidieh, Thomas Hartvigsen, Frank Rudzicz, and Marzyeh Ghassemi. 2022. The Road to Explainability is Paved with Bias: Measuring the Fairness of Explanations. *arXiv preprint arXiv:2205.03295* (2022).

[11] Emily M. Bender and Batya Friedman. 2018. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. *Transactions of the Association for Computational Linguistics* 6 (2018), 587–604. https://doi.org/10.1162/tacl_a_00041

[12] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José M. F. Moura, and Peter Eckersley. 2020. Explainable machine learning in deployment. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*)*. ACM, 648–657.

[13] Angie Boggust, Benjamin Hoover, Arvind Satyanarayan, and Hendrik Strobelt. 2022. Shared Interest: Measuring Human-AI Alignment to Identify Recurring Patterns in Model Behavior. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. ACM, Article 10, 17 pages.

[14] Brandon Carter, Siddhartha Jain, Jonas Mueller, and David Gifford. 2021. Overinterpretation reveals image classification model pathologies. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 15395–15407. https://proceedings.neurips.cc/paper/2021/hash/8217bb4e7fa0541e0f5e04fea764ab91-Abstract.html

[15] Brandon Carter, Jonas Mueller, Siddhartha Jain, and David K. Gifford. 2019. What made you do this? Understanding black-box decisions with sufficient input subsets. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 567–576.

[16] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 839–847.

[17] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N. Balasubramanian. 2018. Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*. IEEE Computer Society, 839–847.

[18] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, Harald Kittler, and Allan Halpern. 2019. Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC). arXiv:1902.03368 [cs.CV]

[19] Anamaria Crisan, Margaret Drouhard, Jesse Vig, and Nazneen Rajani. 2022. Interactive Model Cards: A Human-Centered Approach to Model Documentation. *arXiv preprint arXiv:2205.02894* (2022).

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 248–255.

[21] Li Deng. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine* 29, 6 (2012), 141–142.

[22] Mark Díaz, Ian Kivlichan, Rachel Rosen, Dylan Baker, Razvan Amironesei, Vinodkumar Prabhakaran, and Emily Denton. 2022. Crowdworksheets: Accounting for individual and collective identities underlying crowdsourced dataset annotation. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2342–2351.

[23] Shuoyang Ding and Philipp Koehn. 2021. Evaluating Saliency Methods for Neural Language Models. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. ACM, 5034–5052.

[24] Dumitru Erhan, Y. Bengio, Aaron Courville, and Pascal Vincent. 2009. Visualizing Higher-Layer Features of a Deep Network. *Technical Report, Univeristé de Montréal* (2009).

[25] European Commission. 2018. Transparent information, communication and modalities for the exercise of the rights of the data subject. https://gdpr-info.eu/art-12-gdpr/

[26] Hugging Face. 2022. *Model Card Guidebook*. https://huggingface.co/docs/hub/model-card-guidebook

[27] Huanming Fang and Hui Miao. 2020. *Introducing the Model Card Toolkit for Easier Model Transparency Reporting*. https://modelcards.withgoogle.com/about

[28] Ruth C. Fong and Andrea Vedaldi. 2017. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE, 3449–3457.

[29] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna M. Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for datasets. *Commun. ACM* 64, 12 (2021), 86–92.

[30] Amirata Ghorbani, Abubakar Abid, and James Y. Zou. 2019. Interpretation of Neural Networks Is Fragile. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*. AAAI, 3681–3688.

[31] Tristan Gomez, Thomas Fréour, and Harold Mouchère. 2022. Metrics for Saliency Map Evaluation of Deep Learning Explanation Methods. In *Pattern Recognition and Artificial Intelligence ICPRAI (Lecture Notes in Computer Science, Vol. 13363)*, Mounim A. El-Yacoubi, Eric Granger, Pong Chi Yuen, Umapada Pal, and Nicole Vincent (Eds.). Springer, 84–95.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[33] Sarah Holland, Ahmed Hosny, Sarah Newman, Joshua Joseph, and Kasia Chmielinski. 2020. The dataset nutrition label. *Data Protection and Privacy, Volume 12: Data Protection and Democracy* 12 (2020), 1.

[34] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2019. A Benchmark for Interpretability Methods in Deep Neural Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 9734–9745.

[35] Ben Hutchinson, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell. 2021. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 560–575.

[36] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn L. Ball, Katie S. Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. 2019. CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*. AAAI, 590–597.

[37] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Xiaoguang Han. 2020. Adv-watermark: A Novel Watermark Perturbation for Adversarial Examples. In *Proceedings of the International Conference on Multimedia*. ACM, 1579–1587.

[38] Hyungsik Jung and Youngrock Oh. 2021. Towards Better Explanations of Class Activation Mapping. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 1316–1324. https://doi.org/10.1109/ICCV48922.2021.00137

[39] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda B. Viégas, and Michael Terry. 2019. XRAI: Better Attributions Through Regions. In *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE, 4947–4956.

[40] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. 2019. The (Un)reliability of Saliency Methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Vol. 11700. Springer, 267–280.

[41] Pieter-Jan Kindermans, Kristof T Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. 2017. Learning how to explain neural networks: Patternnet and patternattribution. *arXiv preprint arXiv:1705.05598* (2017).

[42] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. 2020. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896* (2020).

[43] Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. 2022. The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective. *arXiv preprint arXiv:2202.01602* (2022).

[44] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. 2015. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the Conference on Intelligent User Interfaces (IUI)*. 126–137.

[45] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. *Advances in neural information processing systems* 30 (2017).

[46] Heyi Li, Yunke Tian, Klaus Mueller, and Xin Chen. 2019. Beyond saliency: understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation. *Image and Vision Computing* 83 (2019), 70–86.

[47] Xiao-Hui Li, Yuhan Shi, Haoyang Li, Wei Bai, Caleb Chen Cao, and Lei Chen. 2021. An Experimental Study of Quantitative Evaluations on Saliency Methods. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 3200–3208.

[48] Ge Liu, Haoyang Zeng, Jonas Mueller, Brandon Carter, Ziheng Wang, Jonas Schilz, Geraldine Horny, Michael E Birnbaum, Stefan Ewert, and David K Gifford. 2020. Antibody complementarity determining region design using high-capacity machine learning. *Bioinformatics* 36, 7 (2020), 2126–2133.

[49] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*. 4765–4774.

[50] Angelina McMillan-Major, Salomey Osei, Juan Diego Rodriguez, Pawan Sasanka Ammanamanchi, Sebastian Gehrmann, and Yacine Jernite. 2021. Reusable templates and guides for documenting datasets and models for natural language processing and generation: A case study of the HuggingFace and GEM data and model cards. *arXiv preprint arXiv:2108.07374* (2021).

[51] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*)*. ACM, 220–229.

[52] Christoph Molnar. 2019. *Interpretable Machine Learning*.

[53] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition* 65 (2017), 211–222.

[54] Mohammed Bany Muhammad and Mohammed Yeasin. 2020. Eigen-CAM: Class Activation Map using Principal Components. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*. IEEE, 1–7.

[55] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. 2018. The Building Blocks of Interpretability. *Distill* (2018). https://doi.org/10.23915/distill.00010 https://distill.pub/2018/building-blocks.

[56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NIPS)* 32 (2019).

[57] Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA, 151.

[58] Jacob Pfau, Albert T. Young, Maria L. Wei, and Michael J. Keiser. 2019. Global Saliency: Aggregating Saliency Maps to Assess Dataset Artefact Bias. https://doi.org/10.48550/ARXIV.1910.07604

[59] Mihaela Porumb, Saverio Stranges, Antonio Pescapè, and Leandro Pecchia. 2020. Precision medicine and artificial intelligence: a pilot study on deep learning for hypoglycemic events detection based on ECG. *Scientific reports* 10, 1 (2020), 1–16.

[60] Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. 2022. Data Cards: Purposeful and Transparent Dataset Documentation for Responsible AI. *arXiv preprint arXiv:2204.01075* (2022).

[61] Harish Guruprasad Ramaswamy et al. 2020. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 983–991.

[62] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 1135–1144.

[63] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2017. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Transactions Neural Networks and Learning Systems* 28, 11 (2017), 2660–2673.

[64] Adriel Saporta, Xiaotong Gui, Ashwin Agrawal, Anuj Pareek, Steven QH Truong, Chanh DT Nguyen, Van-Doan Ngo, Jayne Seekins, Francis G Blankenberg, Andrew Y Ng, et al. 2022. Benchmarking saliency methods for chest X-ray interpretation. *Nature Machine Intelligence* 4, 10 (2022), 867–878.

[65] Christin Seifert, Stefanie Scherzinger, and Lena Wiese. 2019. Towards generating consumer labels for machine learning models. In *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*. IEEE, 173–179.

[66] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE, 618–626.

[67] Hong Shen, Wesley H Deng, Aditi Chattopadhyay, Zhiwei Steven Wu, Xu Wang, and Haiyi Zhu. 2021. Value cards: An educational toolkit for teaching social impacts of machine learning through deliberation. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 850–861.

[68] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *Proceedings of the International Conference on Machine Learning (ICML)*, Vol. 70. PMLR, 3145–3153.

[69] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Proceedings of the International Conference on Learning Representations (ICLR), Workshop Track*.

[70] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. SmoothGrad: Removing noise by adding noise. *arXiv preprint arXiv:1706.03825* (2017).

[71] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. 2015. Striving for Simplicity: The All Convolutional Net. In *Proceedings of the International Conference on Learning Representations (ICLR), Workshop Track*.

[72] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 3319–3328.

[73] Mukund Sundararajan, Jinhua Xu, Ankur Taly, Rory Sayres, and Amir Najmi. 2019. Exploring Principled Visualizations for Deep Network Attributions. In *Proceedings of the Conference on Intelligent User Interfaces (IUI) Workshops (CEUR Workshop Proceedings, Vol. 2327)*.

[74] Harini Suresh, Steven R Gomez, Kevin K Nam, and Arvind Satyanarayan. 2021. Beyond expertise and roles: A framework to characterize the stakeholders of interpretable machine learning and their needs. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–16.

[75] Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun D. Preece. 2020. Sanity Checks for Saliency Metrics. In *Proceedings of the Conference on Artificial Intelligence*. AAAI, 6021–6029.

[76] Sana Tonekaboni, Shalmali Joshi, Melissa D. McCradden, and Anna Goldenberg. 2019. What Clinicians Want: Contextualizing Explainable Machine Learning for Clinical End Use. In *Proceedings of the Machine Learning for Healthcare Conference (MLHC)*, Vol. 106. PMLR, 359–380.

[77] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. 2020. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 24–25.

[78] Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. 2021. Noise or Signal: The Role of Image Backgrounds in Object Recognition. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=gl3D-xY7wLq

[79] Mengjiao Yang and Been Kim. 2019. Benchmarking attribution methods with relative feature importance. *arXiv preprint arXiv:1907.09701* (2019).

[80] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. 2019. On the (In)fidelity and Sensitivity of Explanations. In *Advances in Neural Information Processing Systems (NeurIPS)*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 10965–10976.

[81] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 8689)*, David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer, 818–833.

[82] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. 2018. Top-down neural attention by excitation backprop. *International Journal of Computer Vision* 126, 10 (2018), 1084–1102.

[83] Jianming Zhang, Zhe L. Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. 2016. Top-Down Neural Attention by Excitation Backprop. In *European Conference on Computer Vision (ECCV)*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.), Vol. 9908. Springer, 543–559.

[84] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning Deep Features for Discriminative Localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2921–2929.

A APPENDIX

A.1 Additional Saliency Cards

The saliency cards repository (https://github.com/mitvis/saliency-cards) contains a saliency card template (Fig. A1) and example saliency cards, including Grad-CAM (Fig. 2) and integrated gradients (Fig. A2). As more saliency methods are documented, developed, and evaluated, we expect the repository to serve as a centralized location for saliency documentation.

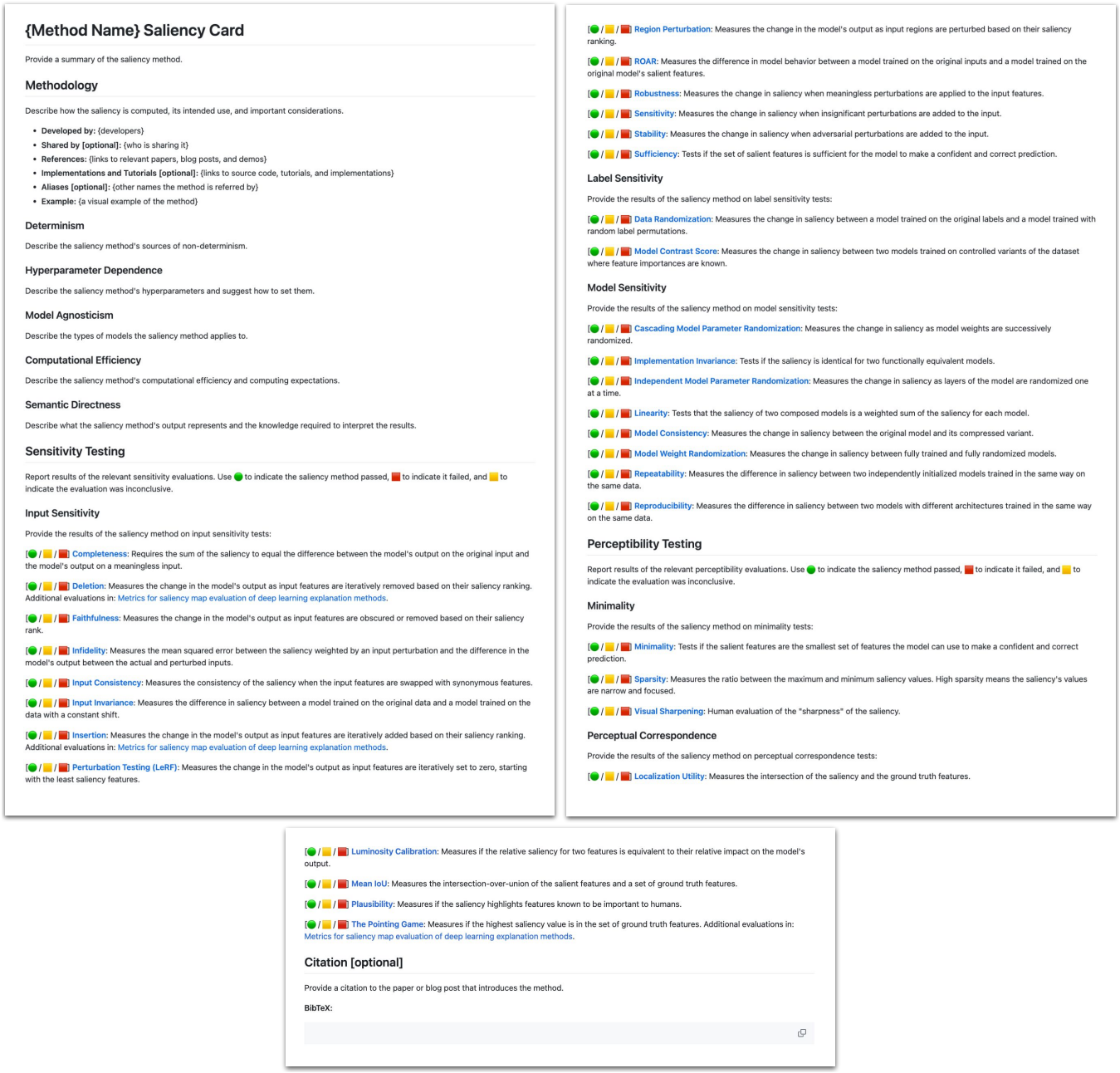


Figure A1: The saliency card template provided at https://github.com/mitvis/saliency-cards. The template describes what to include in each section of the saliency card and summarizes existing evaluations for sensitivity and perceptibility attributes.

Integrated Gradients Saliency Card

Integrated gradients is a model-dependent, path-attribution saliency method.

Methodology

Integrated gradients computes saliency by comparing the saliency of the actual input to the saliency of a meaningless baseline input. It does so by approximating the integral of the gradient of the target output with respect to the input features, linearly interpolating from the baseline to the actual input.

Developed by: Mukund Sundararajan, Ankur Taly, and Qiqi Yan at Google.

References:

- *Original Paper*: Axiomatic Attribution for Deep Networks
- *Paper on Integrated Gradients Hyperparameters*: Visualizing the Impact of Feature Attribution Baselines

Implementations and Tutorials:

- *Original GitHub Repository*: ankurtaly/Integrated-Gradients
- *PyTorch Integration via Captum*: Captum Integrated Gradients
- *TensorFlow Integration*: TensorFlow Integrated Gradients Tutorial

Aliases: Path-Integrated Gradients

Example: The integrated gradients saliency map (right) on an ImageNet image of a `cab` (left) using a Pytorch pretrained ResNet50. This example is computed in `integrated_gradients_example.ipynb`.

Determinism

Integrated gradients is deterministic unless the user chooses a non-deterministic `baseline` value.

Hyperparameter Dependence

Integrated gradients is sensitive to its `baseline` parameter. Since integrated gradients computes feature importance by integrating from a meaningless `baseline` to the actual input, its saliency is zero for any features where the `baseline` and input values are the same.

The all-zero `baseline` is common; however, other options include random noise, a blurred input, the inverse of the input, the input with added noise, or the average of multiple baselines. For more information on the `baseline` parameter and suggestions for how to set it, see: Visualizing the Impact of Feature Attribution Baselines

Model Agnosticism

Integrated gradients requires a differentiable model with access to the gradients.

Computational Efficiency

Computing integrated gradients takes on the order of 1e-1 seconds using the Captum implementation on a 224x224x3 dimensional ImageNet image, ResNet50 model, and one NVidia G100 GPU.

Semantic Directness

The output of integrated gradients is the accumulated gradient between the `baseline` input and the actual input. Interpreting its output requires understanding model gradients and the impact of the `baseline` hyperparameter.

Sensitivity Testing

Input Sensitivity

🟢 **Completeness**: Integrated gradients algorithmically guarantees completeness. The sum of the integrated gradients will equal the difference in the model's output between the actual and `baseline` inputs.

🟨 **Infidelity**: Integrated gradients' infidelity is inconclusive. Integrated gradients outperforms vanilla gradients on MNIST and ImageNet, performs equivalently to vanilla gradients on CIFAR-100, and performs worse than vanilla gradients with SmoothGrad, guided backpropagation, guided backpropagation with SmoothGrad, and integrated gradients with SmoothGrad across all three datasets.

🟨 **Input Consistency**: Integrated gradients' input consistency is inconclusive. It was more consistent than vanilla gradients and SmoothGrad using the LSTM and QRNN models but less consistent than SmoothGrad using a transformer model. Evaluated using number and gender agreement feature swaps with the Syneval and Winobias datasets.

🟨 **Input Invariance**: Integrated gradients can fail input invariance due to its `baseline`. A 0-vector `baseline` is not input invariant, while a black baseline is input invariant. Regardless of the baseline, it is possible to selecting an input transformation that guarantees integrated gradients fails input invariance. Evaluated using a CNN on MNIST.

🟨 **Perturbation Testing (LeRF)**: Integrated gradients' LeRF perturbation tests were inconclusive. It passed for experiments using MNIST, CIFAR-10, and IMDB datasets with MLP, CNN, and LSTM models. However, it had worse than random performance using ImageNet Inception V3. This failure is due to incorrectly estimating the sign of the saliency, causing important features (with negative saliency) to be removed first.

🟢 **Perturbation Testing (MoRF)**: Integrated gradients passes all MoRF perturbation tests. Evaluated using MNIST, CIFAR-10, ImageNet, and IMDB datasets with MLP, CNN, Inception V3, and LSTM models.

🟥 **ROAR**: Integrated gradients fails the ROAR test. It performs worse than random saliency when applied to an ImageNet ResNet50.

🟨 **Robustness**: Integrated gradients is somewhat sensitive to random noise, which causes slight changes to the saliency. However, integrated gradients outperforms vanilla gradients, input x gradient, LRP, Occlusion, and LIME. Evaluated on MNIST CNNs.

🟨 **Sensitivity**: Integrated gradients' sensitivity tests are inconclusive. It is less sensitive to meaningless perturbations than vanilla gradients and guided backpropagation but more sensitive than SmoothGrad applied to vanilla gradients, integrated gradients, and guided backpropagation. Evaluated on MNIST, CIFAR-10, and ImageNet.

🟢 **Stability**: Integrated gradients is relatively stable, and its outputs change minimally in response to adversarial perturbations. It performs better than LIME and equivalently to vanilla gradients, input x gradients, LRP, and Occlusion. Evaluated on MNIST.

Label Sensitivity

🟨 **Data Randomization**: Integrated gradients changes appropriately when the model is trained on perturbed data labels. However, its visualizations can misleadingly show input structure. Evaluated on MNIST and Fashion MNIST.

🟢 **Model Contrast Score**: Integrated gradients achieves a better-than-random model contrast score but performs worse than Grad-CAM and SmoothGrad. Evaluated on the BAM image dataset.

Model Sensitivity

🟨 **Cascading Model Parameter Randomization**: Integrated gradients changes appropriately as the model is progressively randomized. However, its visualizations can misleadingly show input structure. Evaluated an ImageNet Inception V3.

🟢 **Implementation Invariance**: Integrated gradients algorithmically guarantees implementation invariance. It will produce equivalent saliency for functionally equivalent models.

🟨 **Independent Model Parameter Randomization**: Integrated gradients changes appropriately as the model layers are independently randomized. However, its visualizations can misleadingly show input structure. Evaluated an ImageNet Inception V3.

🟢 **Linearity**: Integrated gradients algorithmically guarantees linearity. Its saliency on a model composed of two sub-models will equal the weighted sum of its saliency on each sub-model.

🟨 **Model Consistency**: Integrated gradients' model consistency is inconclusive. In response to model compression, its saliency stayed more consistent than vanilla gradients and SmoothGrad on LSTM and QRNN models but was less consistent than SmoothGrad on a transformer model. Evaluated using DistillBert distillation and number and gender agreement tasks on the Syneval and Winobias datasets.

🟨 **Model Weight Randomization**: On a randomized model, integrated gradients saliency is near-random. Evaluated on SIIM-ACR Pneumothorax and RSNA Pneumonia medical images.

🟥 **Repeatability**: Integrated gradients fails repeatability. Its saliency values for two models trained in the same way are more dissimilar than similar. Evaluated on SIIM-ACR Pneumothorax and RSNA Pneumonia medical images.

🟥 **Reproducibility**: Integrated gradients fails reproducibility. Its saliency values for two different architectures trained in the same way are more dissimilar than similar. Evaluated using Inception V3 and DenseNet-121 on SIIM-ACR Pneumothorax and RSNA Pneumonia medical images.

Perceptibility Testing

Minimality

🟥 **Visual Sharpness**: Integrated gradients are less visually coherent than SmoothGrad saliency methods. Evaluated on an ImageNet Inception V3 and MNIST CNN.

Perceptual Correspondence

🟥 **Localization Utility**: Integrated gradients fails localization utility. Its saliency values overlap less with the ground truth than a random model. Evaluated on SIIM-ACR Pneumothorax and RSNA Pneumonia medical images.

🟥 **Mean IoU**: Integrated gradients saliency has lower mean IoU with the ground truth features than CAM and occlusion-based saliency method and human localization. Evaluated using CNNs on CheXpert chest x-ray images.

🟢 **Plausibility**: Integrated gradients highlights human-important features more often than vanilla gradients and equivalently to SmoothGrad on LSTM, QRNN, and transformer models. Evaluated on number and gender agreement tasks using the Syneval and Winobias datasets.

🟨 **The Pointing Game**: Integrated gradient's most salient feature in the ground truth region as many times as other saliency methods but less than human localization. Evaluated using CNNs on CheXpert chest x-ray images by Benchmarking saliency methods for chest X-ray interpretation.

Citation

```
@inproceedings{integratedgradients
  author    = {Mukund Sundararajan and Ankur Taly and Qiqi Yan},
  title     = {Axiomatic Attribution for Deep Networks},
  booktitle = {International Conference on Machine Learning ({ICML})},
  publisher = {{PMLR}},
  year      = {2017},
}
```

Figure A2: The saliency card for integrated gradients [72] provided at https://github.com/mitvis/saliency-cards.

A.2 Saliency Card Attribute Comparison

Saliency cards expose differences in the methodology, sensitivity, and perceptibility of saliency methods. We have provided example saliency cards for Grad-CAM [66] (Fig. 2) and integrated gradients [72] (Fig. A2) that describe their attributes. Here, we compare the saliency card attributes for additional saliency methods, to analyze the research landscape and reveal gaps and opprotunities for future work.

A.2.1 Methodology Comparison. In Table A1, we compare the methodological attributes of 11 saliency methods. We extract each method's determinism, hyperparameter dependence, model agnosticism, and semantic directness from its original paper and we compute each method's computational efficiency.

We compute computational efficiency of each method on a 224x224x3 ImageNet [20] image and a pretrained PyTorch [56] ResNet50 [32] using one Nvidia G100 GPU. For consistency, we use each method's default parameters. To account for noise in our timing procedure, we report the order of magnitude of the computation time as opposed to the raw value. The results of the computational efficiency tests can be found at: https://github.com/mitvis/saliency-cards/blob/main/paper_results/computational_efficiency.ipynb.

For vanilla gradients [24, 69], guided backprop [71], Grad-CAM [66], integrated gradients [72], input x gradient, and SHAP [49], we use their Captum [42] implementations. For SmoothGrad [70] and XRAI [39], we use the public implementation from Google PAIR[2]. For RISE [3] [57], SIS [15] [4], and LIME [5] [62], we use the public implementations provided by their authors. To provide a consistent saliency method interface across the different implementations, we provide a wrapper for each method and visualization code at: https://github.com/mitvis/saliency-cards.

This is just one way to test the computational efficiency of each saliency method. We expect the relative computational efficiencies may change depending on the implementation, model architecture, parameter settings, and data modality. Similarly, there may be theoretical computational efficiency bounds that can be derived for each method. As additional experiments reveal new computational efficiency results, saliency cards can evolve to contain a comprehensive overview of each method's computational efficiency.

A.2.2 Sensitivity and Perceptibility Comparison. Table A2 summarizes the results of 33 saliency method evaluations that span the sensitivity and perceptibility attributes. We categorize each test's results for every saliency method it tested. Based on what each evaluation paper reports, we group the results into "pass", "fail", and "inconclusive".

While we try to directly report the paper's claims about each saliency method, in some cases, these categorizations can be subjective.

Table A1: We compare the methodological attributes of 11 saliency methods to systematically analyze the research landscape to understand gaps and opportunities for future work.

	Determinism	Hyperparameter Dependence	Model Agnosticism	Computational Efficiency	Semantic Directness
Vanilla Gradients [24, 69]	Deterministic.	None.	Requires a differentiable model with access to gradients.	On the order of 1e−2 seconds.	The magnitude of the change in the model's output given a small change to an input feature.
SmoothGrad [70]	Non-deterministic noise perturbations.	Gaussian noise parameters; the number of samples to average over.	Applicable to any saliency method.	Adds a ∼20x time increase.	The average saliency across noisy versions of the input.
Guided BackProp [71]	Deterministic unless using a non-deterministic saliency method.	Saliency method (typically vanilla gradients).	Requires a differentiable model with access to gradients.	On the order of 1e−2 seconds using vanilla gradients.	The output of another gradient-based saliency method only considering paths through the model with positive gradients.
Grad-CAM [66]	Deterministic.	Interpolation method to upsample with; choice of convolutional layer (typically the last convolutional layer).	Requires a differentiable model, access to the gradients, and a convolutional layer.	On the order of 1e−2 seconds.	The positive attributions of the gradient-weighted feature maps from an internal convolutional layer.
Integrated Gradients [72]	Deterministic unless using a non-deterministic baseline.	Baseline value; integral approximation parameters.	Requires a differentiable model with access to gradients.	On the order of 1e−1 seconds.	The accumulated gradient between the baseline input and the actual input.
Input X Gradient	Deterministic.	None.	Requires a differentiable model and access to gradients.	On the order of 1e−2 seconds.	The input feature value weighted by the gradient.
XRAI [39]	Deterministic unless using a non-deterministic saliency method or segmentation method.	Segmentation method; saliency method (typically integrated gradients).	Requires input features that can be meaningfully clustered (e.g., image pixels).	On the order of 1e1 seconds.	The input regions with the largest sum of feature attribution.
RISE [57]	Non-deterministic mask generation.	Masking value; mask generation parameters.	No requirements on the model or access to internals.	On the order of 1e−1 seconds.	The sum of input masks weighed by the model's confidence on the masked input.
SIS [15]	Deterministically produces a set of explanations per input.	Feature replacement values; model confidence threshold.	No requirements on the model or access to internals.	SIS: prohibitively slow. Batched Gradient SIS: On the order of 1e1 seconds.	The minimum set of pixels necessary for the model to confidently produce the same output.
LIME [62]	Non-deterministic perturbations.	Linear surrogate model and parameters; input perturbation parameters.	No requirements on the model or access to internals.	On the order of 1e1 seconds.	The positively contributing features learned by a surrogate model trained to mimic the original model's local decision boundary for the input.
SHAP [49]	Non-deterministic coalition sampling.	Feature replacement values; linear model parameterization; regularization parameter.	No requirements on the model or access to internals.	GradientSHAP: on the order of 1e−1 seconds. KernelSHAP: on the order of 1e1 seconds.	The impact of each input feature on the output as defined by Shapley values.

[2]https://github.com/PAIR-code/saliency
[3]https://github.com/eclique/RISE
[4]https://github.com/gifford-lab/overinterpretation/tree/master; https://github.com/google-research/google-research/blob/master/sufficient_input_subsets/sis.py
[5]https://github.com/marcotcr/lime

For instance, if an evaluation compares two saliency methods and one outperforms the other, it can be hard to evaluate which methods pass or fail the test. It could be just the top-performing one passes, both pass, or both fail. In evaluations that provide negative and positive controls (e.g., random baselines), we categorize each method based on those controls.

These categorizations are helpful for analyzing the gaps in the design space (e.g., limited evaluations for particular models or attributes); however, saliency cards must also provide a textual summary of the evaluation's results. This description helps mitigate the subjectivity of pass/fail/inconclusive categorizations and can provide helpful context to a user. For instance, seeing a lack of experimental controls could inspire a user to run additional baseline evaluations for an existing test or design new metrics that better separate the behavior of the saliency methods.

Table A2: We summarize the results of 33 saliency method evaluations that describe the saliency card sensitivity and perceptibility attributes. This summarization reveals the need for additional evaluation using existing metrics and new metrics to further test under-evaluated attributes. We generalize the results of each test based on if they passed the evaluation (green ✓), failed the evaluation (red ✗), performed inconclusively (yellow —), or were not tested (grey cell).

		Vanilla Gradients (VG) [24, 69]	VG + SmoothGrad [70]	Guided BackProp (GBP) [71]	Grad-CAM [66]	Guided Grad-CAM [66]	Integrated Gradients (IG) [72]	IG + SmoothGrad [70, 72]	Gradient · Input	XRAI [39]	DeepLift [68]	LRP [46]	Ablation CAM [61]	Score-CAM [77]	RISE [57]	Grad-CAM++ [16]	SIS [15]	Deconvnet [81]	CAM [84]	Occlusion [81]	Eigen-CAM [54]	LIME [62]	GBP + SmoothGrad [70, 71]	SHAP [49]	Deep Taylor Decomposition [53]	PatternNet [41]
Input Sensitivity	Completeness [72]						✓				✓	✓														
	Deletion [57]				—										✓								—				
	Faithfulness [4]																						✓		✓		
	Infidelity [80]	—	✓	—			—	✓																✓	✓		
	Input Consistency [23]	—	—				—																				
	Input Invariance [40]	✓	✓	✓			—	✓	✗			✗												✓		—	✓
	Insertion [57]				—										✓								—				
	Perturbation Testing (LeRF) [6]	—					—		—		—	—								—							
	Perturbation Testing (MoRF) [6]	—					✓		✓		✓	—								✓							
	Region Perturbation [63]											✓						✓									
	ROAR [34]	✗	—	✗			✗																				
	Robustness [3]	—					—		—			—								✗		✗		✗			
	Sensitivity [80]	—	✓	—			—	✓																✓	✓		
	Stability [4]	✓					✓		✓			✓								✓		✗		✗			
	Sufficiency [15]																✓										
Label Sensitivity	Data Randomization [1]	✓	✓	—	✓	—	—	—	—																		
	Model Contrast Score [79]	—	—	—	✓	—	✓	—	—																		
Model Sensitivity	Cascading Model Randomization [1]	✓	✓	✗	✓	✗	—	—	—																		
	Implementation Invariance [72]	✓					✓				✗	✗															
	Independent Model Randomization [1]	✓	✓	✗	✓	✗	—	—	—																		
	Linearity [72]						✓																				
	Model Consistency [23]	—	—				—																				
	Model Weight Randomization [9]	✓	✓	✓	✓	—	—	✓		✗																	
	Repeatability [9]	✗	✗	—	—	—	✗	✗		✓																	
	Reproducibility [9]	✗	✗	✗	—	—	✗	✗		✓																	
Minimality	Minimality [15]																✓										
	Sparsity [31]				—								✓	—	—	—											
	Visual Sharpening [70]	✗	✓	—			✗	✓																✓			
Perceptual Correspondence	Localization Utility [9]	✗	✗	✗	✗	✗	✗	✗		✓																	
	Luminosity Calibration [31]				✗								✗	✗	✓	✗											
	Mean IoU [64]				—	—	✗				✗	✗								—	—						
	Plausibility [23]	—	✓				✓																				
	The Pointing Game [83] / Hit Rate [64]	✓			—		—				—	—				—		✓	✓	✗	—					

A.3 Additional Interview Details

In each interview, we explained saliency card attributes via a definition and example that demonstrated the attribute. We showed U1–U8 examples from ImageNet [20], melanoma classification [18], MNIST digit recognition [21], and CheXpert chest x-rays [36], shown in Fig. A4. Since the radiologist participant (U9) was unfamiliar with machine learning, we only showed them examples using CheXpert chest x-rays [36], shown in Fig. A3.

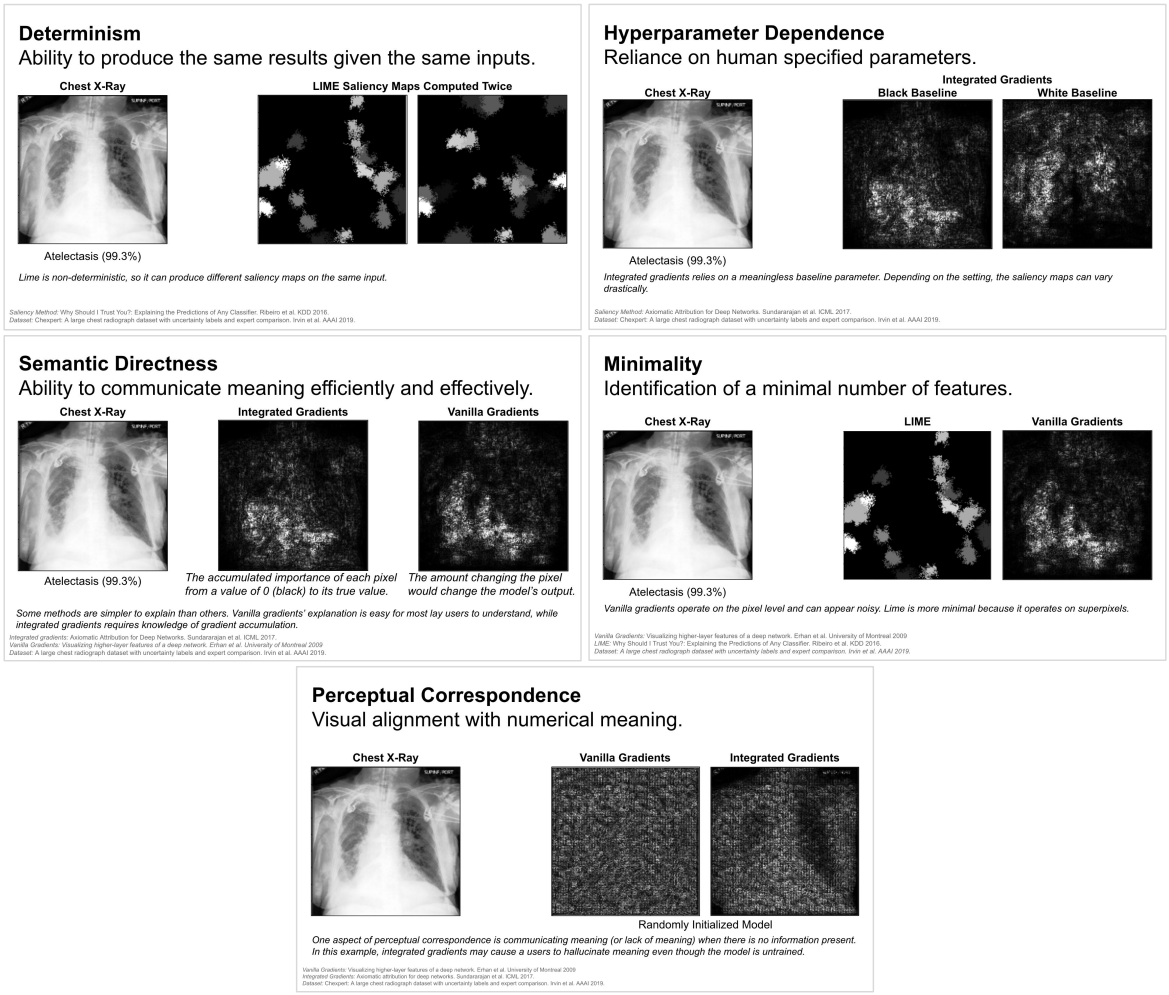


Figure A3: Examples of each saliency card attribute we discussed with the radiologist user (U8) in our user study. Given they were unfamiliar with machine learning, we used medical imaging examples from CheXpert [36]. Each example defines the attribute and shows an informative instance of the attribute exhibited.

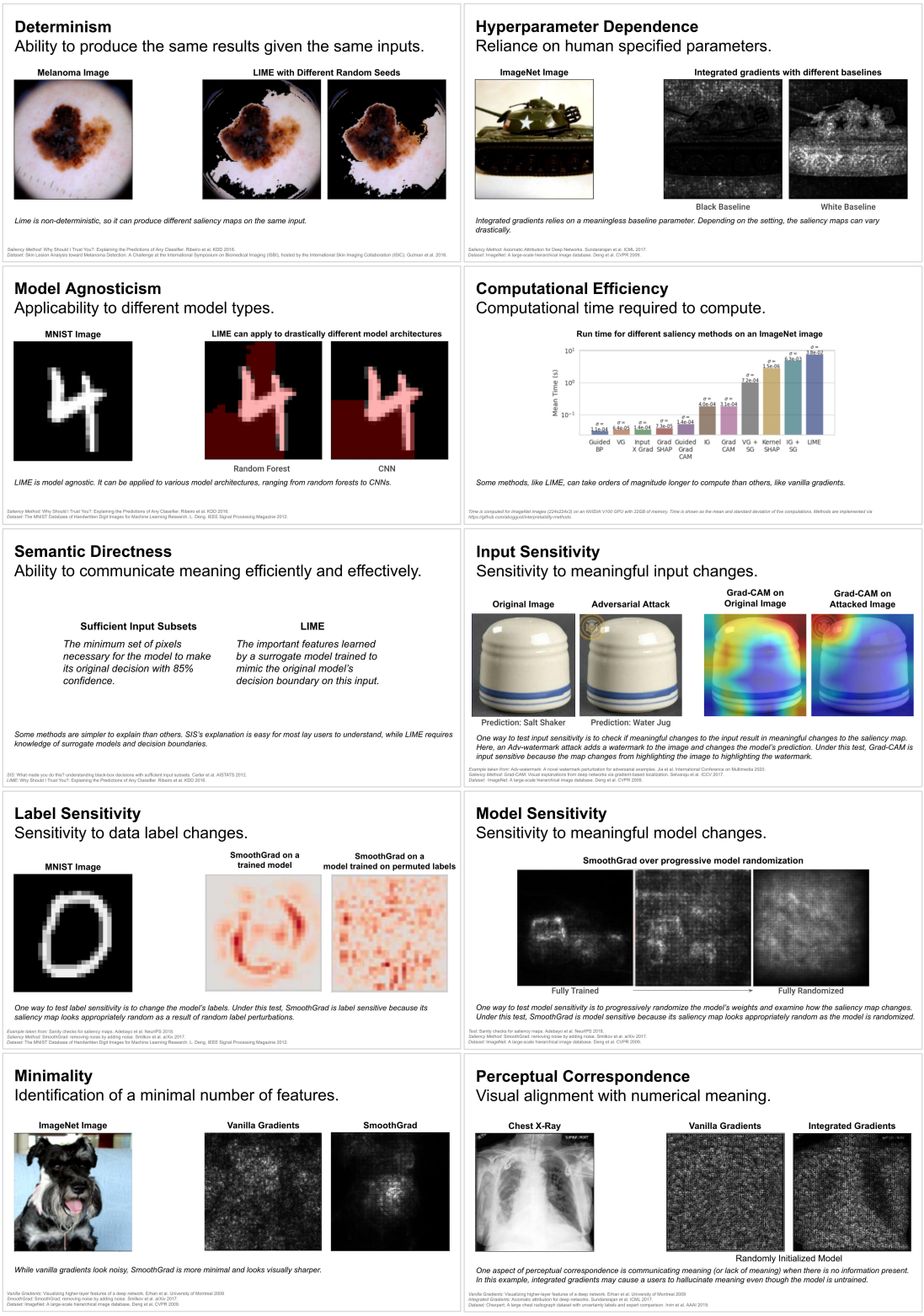


Figure A4: Examples of each saliency card attribute we discussed with the U1–U7. Each example defines the attribute and shows an informative instance of the attribute exhibited.