Interpretable Artificial Intelligence: A Survey of Methods and Applications

Peizhuo Liu

High School attatched to Northwestern Polytechnical University

October 17, 2024

Abstract

Interpretable artificial intelligence (XAI) has become a critical area of research in the last five years, driven by the need to understand and trust complex machine learning models. This literature review surveys recent theoretical and methodological developments in interpretable AI methods, with an emphasis on prominent model-agnostic explanation techniques such as SHAP (SHapley Additive exPlanations) [33] and LIME (Local Interpretable Model-Agnostic Explanations) [38]. We outline the mathematical formulations of SHAP and LIME and discuss their key theoretical contributions, including the game-theoretic foundations of SHAP and the locally linear surrogate approach of LIME.

Beyond these, we cover a broad spectrum of interpretability methods: intrinsically interpretable models (such as sparse linear models [53], generalized additive models [21], and decision trees [9]), post-hoc explanation methods (encompassing feature attribution, counterfactual explanations [55], rule extraction, and example-based reasoning), visualization-based techniques (notably saliency maps [47] and attention-based explanations for deep networks [7]), and emerging hybrid approaches that integrate interpretability into model design.

We provide an academic analysis of each class of methods, discussing their strengths, limitations, and the trade-offs involved in fidelity, interpretability, and scope (local vs. global). We also compare these approaches to highlight how they complement each other in explaining different aspects of black-box models. The review is structured as a scholarly article with an introduction to the field, detailed sections on each category of methods, and a conclusion summarizing current trends and future directions. All methods are discussed with technical depth and supported by citations from peer-reviewed literature.

Contents

1 Introduction

Machine learning models, especially complex ones like ensemble methods and deep neural networks, often operate as “black boxes” with high predictive power but low interpretability. In high-stakes domains such as healthcare, finance, and autonomous systems, explainable AI (XAI) methods are essential to provide insight into model decisions, ensure accountability, and satisfy regulatory requirements [15, 19].

Key Challenge: As AI systems become more complex and pervasive in critical applications, the inability to understand their decision-making process poses significant challenges for trust, accountability, and regulatory compliance.

Over the past five years, research in interpretable AI has grown exponentially, yielding new methods and theoretical understandings that aim to make machine learning models more transparent without significantly sacrificing accuracy [1]. This literature review synthesizes these developments, focusing on the theoretical foundations and methodological advances in model interpretability.

We organize the discussion into several major categories of interpretability techniques. First, we review intrinsically interpretable models, which are designed to be understandable by humans *by their very structure*. Examples include linear models, decision trees, and rule-based systems. These models trade some accuracy for transparency, and recent work has sought to improve their expressive power while retaining interpretability [41].

Next, we examine post-hoc interpretability methods, which are applied *after* a complex model is trained to explain its predictions. In this category, we highlight two widely used methods, LIME and SHAP, due to their substantial influence on the field. We detail their mathematical formulation and theoretical properties – LIME’s approach of fitting local surrogate models (typically sparse linear models) around a prediction of interest [38] and SHAP’s use of Shapley values from cooperative game theory to allocate credit for a prediction among features [33]. We then broaden our survey to other post-hoc methods, including feature attribution techniques (such as gradient-based saliency maps, integrated gradients [51], and layer-wise relevance propagation [6]), counterfactual and contrastive explanations that answer “what if” questions by perturbing inputs [55], and rule extraction methods that derive human-readable rules or decision sets from complex models [13].

Furthermore, we dedicate a section to visualization-based techniques. These methods often overlap with feature attribution, especially in the context of deep learning for images and text, where saliency maps and activation visualizations help identify which parts of an input are most influential for a model’s prediction. We review methods like Grad-CAM, which produces class-specific heatmaps highlighting important regions in an image [42], and related approaches that enhance interpretability of convolutional networks.

Finally, we explore emerging hybrid approaches that blur the line between interpretable models and post-hoc explanations. These include models with architecture or training objectives specifically designed for interpretability – for example, concept bottleneck models that force an intermediate layer to represent human-understandable concepts [27], and prototype-based networks that reason by comparing inputs to learned prototypical examples [30]. Such approaches seek to achieve high accuracy while baking in interpretability, addressing the critique that post-hoc explanations might sometimes be misaligned with the true model internals [55].

Throughout this review, we discuss the strengths and limitations of each interpretability approach. We emphasize that no single method is universally best – each involves trade-offs between fidelity (accuracy in representing the original model), interpretability (cognitive simplicity for humans), locality (explaining individual predictions vs. global model behavior), and assumptions (e.g., feature independence in SHAP or linearity in LIME's local models). Comparative analyses are provided to illustrate how methods complement one another; for instance, how feature attributions can be combined with counterfactual explanations to give both *how* and *how else* insights into a prediction. All claims are backed by peer-reviewed studies from major conferences and journals to ensure an authoritative survey of the state of the art.

In the following sections, we delve into each category, starting with intrinsically interpretable models and then covering the gamut of post-hoc and hybrid interpretability methods. We conclude with a summary of current trends, open challenges (such as evaluating explanations and ensuring their robustness), and future directions in interpretable AI research.

2 Intrinsically Interpretable Models

Intrinsically interpretable models are those whose internal structure can be directly examined and understood by humans. Unlike black-box models, these models aim to maintain transparency without requiring external explanation tools. In practice, intrinsically interpretable models are often simpler or more constrained in form, which can sometimes come at the cost of predictive performance. Recent research has focused on expanding the flexibility of these models while keeping them understandable. We discuss two primary classes: **(1)** linear and additive models, and **(2)** decision trees and rule-based models.

2.1 Linear and Additive Models

Linear models, such as linear regression and logistic regression, have long been favored for their interpretability. Each feature contributes to the prediction via a weight coefficient, making the effect of each input feature *explicit*. A clinician or policymaker can often reason about the sign and magnitude of these weights. However, simple linearity can be too restrictive for complex tasks.

Generalized Additive Models (GAMs) offer a compromise by modeling the prediction as a sum of feature-wise functions:

$$f(x) = \beta_0 + \sum_i g_i(x_i),$$

where each g_i is a univariate shape function learned from data [23]. This retains interpretability—since each feature's contribution can be visualized as a curve—while capturing *nonlinear* relationships.

Recent extensions of GAMs using boosted trees or neural networks have achieved high accuracy on tabular data with interpretable outputs for each feature. For example, **Explainable Boosting Machines (EBMs)** are GAMs learned via gradient boosting that can match complex models in accuracy while remaining intelligible to users [11].

Moreover, researchers have developed **sparse linear models with discrete weights** for domains like healthcare to ensure simplicity. An example is the **RiskSLIM** approach (Supersparse Linear Integer Models), which produces scorecards—simple additive point systems for decision making (e.g., credit scoring or medical risk scores) [54]. These models constrain weights to be small integers, enabling easy arithmetic for human users. Similarly, **linear models with monotonicity constraints** ensure that predictions move in a direction consistent with domain knowledge—useful in fields like finance and healthcare to guarantee that increasing certain risk factors never decreases the predicted risk [20].

The advantage of linear and additive models is their **global interpretability**: the model as a whole can be understood by inspecting its coefficients or component functions. These models inherently avoid the need for post-hoc explanations—*the explanation is the model*—and thus sidestep concerns about fidelity.

However, their simplicity can also be a limitation: they may underfit complex interactions or fail to capture higher-order feature combinations that black-box models can exploit. To mitigate this, recent research has introduced interpretable interaction models—for instance, **GA^{2}M models** (Generalized Additive Models with pairwise interactions), which include selected interaction terms while preserving graphical interpretability for each pair [31].

Nonetheless, there remains a recognized accuracy-vs-interpretability trade-off: as models become more flexible, they risk losing transparency. A notable viewpoint argues that in high-stakes decision-making, it is preferable to use inherently interpretable models rather than rely on post-hoc explanations of black-boxes. This perspective, championed by Rudin *et al.*[41], has catalyzed efforts to enhance the accuracy and usability of intrinsically interpretable models for critical applications.

2.2 Decision Trees and Rule-Based Models

Decision trees are another classic family of interpretable models. A decision tree splits data along feature values, yielding a flowchart-like structure where each path from root to leaf represents a series of decision rules that lead to a prediction. **Shallow trees** (with limited depth) are especially interpretable: each internal node poses a human-understandable question (e.g., “age $>$ 50?”), and the leaves return a prediction.

Tree models are intrinsically **local-by-design**—each prediction can be explained by the single path of conditions that applied to that instance. Decision trees have seen a resurgence in interpretable ML research, with efforts to optimize them for accuracy under complexity constraints. For example, recent methods learn **optimal decision trees** using mixed-integer programming or advanced heuristics, rather than greedy splitting [8].

Rule lists extend this paradigm: they consist of a prioritized list of if-then rules with a default outcome. Angelino *et al.* [4] introduced a method to learn *certifiably optimal rule lists* using a branch-and-bound algorithm that ensures minimal complexity for a given level of accuracy. These models are easy to interpret—one simply follows the first rule that applies, akin to using a checklist.

Rule-based systems generalize decision trees by relaxing structural constraints. Rules can be an unordered set (**decision sets**) or an ordered list (**decision lists**). In a decision set, multiple rules may apply and conflict resolution is needed, while a decision list processes instances in sequence, using the first matching rule.

Interpretable decision sets have been proposed to optimize for brevity (fewer rules and conditions) and low overlap (each instance matches few rules) [28]. Recent work has also introduced **Bayesian rule list models**, which apply Bayesian priors to favor sparse, interpretable structures and use posterior inference to balance accuracy with simplicity [29].

The strength of trees and rule-based models lies in their **high interpretability and logical structure**. Domain experts often find rules or small trees intuitive, as they closely resemble human decision processes or formal policies. They also naturally handle mixed data types and yield *explanations that are faithful by construction*—the explanation for a prediction is literally the part of the model used to make that prediction.

However, these models can become unwieldy if grown too large. A decision tree with dozens of levels or a rule list with hundreds of rules is effectively as opaque as a neural network. **Pruning techniques** and **complexity regularization**—such as limiting tree depth or the number of rules—are thus essential to maintain interpretability [37, 17].

In practice, complex black-box models like **random forests** or **gradient-boosted trees** (which are ensembles of many trees) are not considered interpretable. While summary importance measures can be extracted from such models, these are often insufficient for detailed, instance-level understanding. As a result, **post-hoc explanation methods** are commonly applied to interpret ensembles, leading to the next part of this review.

In summary, intrinsically interpretable models offer a transparent alternative to black-boxes, and recent advances have improved their accuracy and flexibility. Nonetheless, the need for more expressive models—without sacrificing interpretability—has driven researchers to develop *post-hoc* explanation methods for complex models, which we discuss next.

3 Post-hoc Interpretability Methods

Post-hoc interpretability methods are applied *after* a model has been trained, in order to explain its behavior or individual predictions. These techniques treat the original model as a black-box function $f(x)$, and aim to provide insights without modifying f itself.

Post-hoc techniques can be either **model-agnostic**—making few assumptions about the structure of f and applicable to any classifier or regressor—or **model-specific**, which leverage internal details such as gradients in neural networks or tree structures in ensembles for more efficient explanations [1].

This section covers several major classes of post-hoc methods: **surrogate models** (with a focus on LIME and its variants), **Shapley-value-based explanations** (e.g., SHAP), **counterfactual explanations**, and **example-based reasoning** (including prototypes and influential examples). Visualization-based techniques like saliency maps will be covered in the following section.

3.1 Surrogate Models and Local Explanations (LIME and Extensions)

One intuitive approach to explain an individual prediction is to approximate the complex model $f(x)$ in the vicinity of the instance of interest using a simpler, interpretable model.

Local Interpretable Model-Agnostic Explanations (LIME) popularized this approach [38]. LIME generates a synthetic neighborhood around a data point x (typically via random sampling and perturbations of x's features), queries the black-box model to obtain predictions $f(x')$ for each perturbed sample x', and fits a local surrogate model g (such as a sparse linear model or small decision tree) to mimic f in that neighborhood. The explanation is then derived from the structure of g, under the assumption that it faithfully approximates f locally.

To ensure the surrogate model focuses on the locality of x, LIME employs a **locality-weighting function** $\pi_x(x')$ that assigns higher weight to samples x' closer to x, typically using a distance metric such as cosine or Euclidean distance [38]. Formally, LIME seeks a model g from an interpretable class $\mathcal{G}$ (e.g., sparse linear models or shallow decision trees) that minimizes the following objective:

$$\mathcal{L}(f, g, \pi_x) = \sum_{x'} \pi_x(x') \big(f(x') - g(x')\big)^2 + \Omega(g), \tag{1}$$

where $\mathcal{L}$ measures local fidelity to the original model f, and $\Omega(g)$ is a complexity penalty that enforces interpretability (e.g., sparsity or low tree depth).

The resulting explanation model g (e.g., a small set of weighted features) approximates f's behavior in the neighborhood of x. LIME then presents the coefficients of g as the explanation for $f(x)$, often visualized as a list of the most influential features with positive or negative contributions.

LIME is **model-agnostic** and highly flexible. It can be applied to classifiers or regressors across various data modalities—including tabular data, text (by perturbing words or characters), and images (by perturbing superpixels)—as long as a meaningful locality metric can be defined. LIME's explanations are **locally faithful by design**—they aim to approximate f within a neighborhood around x, but are not intended to generalize globally.

However, LIME has notable **limitations**. Since it relies on random sampling to generate perturbed samples, its output can be unstable—multiple runs on the same input may yield different explanations. In addition, the choice of kernel function π_x (i.e., the distance metric and bandwidth) and the perturbation distribution is often heuristic and dataset-dependent. Poor choices may lead LIME to overlook important features or produce misleading explanations [18].

Garreau and von Luxburg [18] provided the first theoretical analysis of LIME, deriving closed-form solutions in the case of linear ground-truth functions and examining LIME's behavior as the sampling density grows large. They showed that in the limit, LIME's explanation converges to the gradient of the true function f at x—a reassuring result that connects LIME to classical sensitivity analysis.

However, they also highlighted a key vulnerability: if the kernel width used in the locality-weighting function $\pi_x(x')$ is not well-tuned to the scale of feature effects, LIME may capture spurious signals or overlook relevant features. A mitigation strategy is to use a robust interpretable model class (e.g., regularized linear models) and to aggregate explanations across multiple LIME runs to reduce variance.

Several extensions to LIME have been proposed to address such limitations. For example, **Stable LIME** and **DLIME (Deterministic LIME)** aim to reduce randomness using deterministic sampling strategies or clustering-based perturbations [1].

Ribeiro *et al.* later introduced **Anchor Explanations** as a high-precision alternative to LIME [39]. Anchors are decision-rule-based explanations: each anchor is a small `if-then` rule—typically involving a condition on one or a few features—that *anchors* the model's prediction.

For instance, in a text classification task, an anchor might be the presence of a specific word that causes the model to predict a particular class, regardless of changes to other parts of the input. Anchors are identified via a search procedure that maximizes **precision**—defined as the probability that the model's prediction remains unchanged when the anchor condition holds and all other features are perturbed.

The result is a human-friendly explanation in logical format, such as: `IF feature A and B are present, THEN prediction is most likely Y`. Compared to coefficient-weighted feature lists, anchors offer clearer decision rules with stronger guarantees of prediction stability under localized perturbation.

In summary, local surrogate methods like LIME and Anchors provide **interpretable, sparse explanations** that excel at answering *"Why did the model predict this?"* in terms of local evidence. Their weaknesses include: No global view of model behavior; Fidelity is inherently local – misleading if assumed to be general; Trust in the surrogate model is essential; visualizations (e.g., local R^2) are often used to assess fit.

Despite their issues, LIME and its variants remain a cornerstone of the XAI toolkit, widely adopted and continuously refined.

3.2 Shapley-Value-Based Explanations: SHAP and Beyond

Another major class of post-hoc explanation methods leverages concepts from cooperative game theory to attribute a model's prediction to individual input features in a *principled and fair* manner.

SHAP (SHapley Additive exPlanations) is a unified framework that connects several earlier feature attribution techniques to the concept of *Shapley values* from cooperative game theory [33].

In game theory, the Shapley value provides a fair way to distribute the total gain achieved by a coalition of players among the individuals, satisfying several axioms of fairness [45]. In the context of machine learning, the "gain" is typically defined as the model output $f(x)$, or the difference from a baseline prediction $\mathbb{E}[f(x)]$, and the "players" are the input features.

The **Shapley value for feature** i is defined as the average marginal contribution of that feature over all possible subsets of the remaining features:

$$\phi_i = \sum_{S \subseteq \mathcal{F} \setminus \{i\}} \frac{|S|!(|\mathcal{F}| - |S| - 1)!}{|\mathcal{F}|!} \big(f(x_{S \cup \{i\}}) - f(x_S)\big) \tag{2}$$

where $\mathcal{F}$ is the full set of features, S is a subset not containing feature i, and x_S is the instance x restricted to the features in S. Intuitively, this measures how much adding feature i to a coalition S changes the prediction, then averages over all such coalitions with fairness-weighted importance.

Shapley values satisfy several desirable axioms:

- **Efficiency:** $\sum_i \phi_i = f(x) - f(\text{baseline})$;
- **Symmetry:** If two features contribute equally in all subsets, they receive equal attributions;
- **Dummy:** Features with no marginal contribution receive zero attribution;
- **Additivity:** Explanations are additive across models: if $f = f_1 + f_2$, then $\phi_i(f) = \phi_i(f_1) + \phi_i(f_2)$.

Lundberg and Lee [33] demonstrated that any additive feature attribution method satisfying a subset of these axioms is mathematically equivalent to computing Shapley values. SHAP is thus appealing for its strong theoretical grounding, offering a principled and *fair* allocation of feature importance.

Computational Challenges. Computing exact Shapley values requires evaluating all $2^{|\mathcal{F}|}$ feature subsets, which becomes computationally infeasible as the number of features grows. To address this, several approximation and optimization strategies have been proposed.

One prominent method is **Kernel SHAP**, which approximates Shapley values by framing the attribution task as a weighted linear regression over randomly sampled feature coalitions [33]. Kernel SHAP is *model-agnostic* and thus widely applicable, but it can be computationally expensive—especially when the model function f is costly to evaluate or the feature set is large.

In contrast, **Tree SHAP** provides a fast and exact algorithm specifically tailored for tree-based models [32]. It leverages dynamic programming to compute attributions efficiently, making it especially suitable for ensemble methods like XGBoost and Random Forests. Tree SHAP guarantees two desirable properties: **consistency**, meaning that increasing a feature's impact in the model never decreases its attribution; and **local accuracy**, meaning that the sum of feature attributions matches the prediction difference from the baseline. These make Tree SHAP more theoretically grounded than traditional measures such as Gini or permutation importance, which often ignore feature interactions or correlations.

Relationship to Other Methods. SHAP serves as a *unifying framework* for model explanation. Several existing techniques can be viewed as special cases of SHAP. For instance, LIME approximates SHAP under specific kernels and sampling strategies. Similarly, **DeepLIFT**, originally developed for neural network interpretation, has been incorporated into the SHAP framework as **DeepSHAP**. SHAP has also been extended to include **SHAP Interaction Values**, which enable attribution not only to individual features but also to pairwise feature interactions.

Limitations. Despite its strong theoretical underpinnings, SHAP is not without limitations. One major issue is the dependence on the choice of **baseline** x', which serves as the reference point for attributions. Poorly chosen baselines may yield unintuitive or misleading results. Additionally, SHAP often assumes **feature independence** when marginalizing over coalitions, which is problematic in the presence of correlated features and can distort attributions.

While **Tree SHAP** provides efficient computation for tree-based models, **Kernel SHAP** remains computationally intensive for high-dimensional data or expensive black-box models.

To address these issues, several extensions have been proposed. **Conditional SHAP** and **causal SHAP** aim to better handle feature dependencies by conditioning on observed values when computing marginal contributions. Other efforts explore alternative axiom systems, relaxing some of SHAP's foundational assumptions to produce new attribution methods with different fairness or efficiency trade-offs [50].

Practical Use. SHAP has become widely adopted in both academic research and industry practice due to its ability to generate **additive explanations**, where the sum of feature attributions $\sum_i \phi_i$ equals the difference between the model's prediction $f(x)$ and a baseline prediction $f(\text{baseline})$ [33]. This additive structure enables intuitive visualizations such as **force plots**, **waterfall charts**, and **summary beeswarm plots**.

Moreover, SHAP supports both **local** and **global** interpretability. On a local level, SHAP values explain individual predictions by quantifying the contribution of each feature. On a global level, aggregating SHAP values across many instances reveals feature importance trends and interaction patterns.

The availability of robust tooling—especially the open-source `shap` Python package—has further fueled SHAP's popularity. With just a few lines of code, practitioners can apply SHAP to models ranging from XGBoost and LightGBM to neural networks and ensemble pipelines.

In essence, SHAP answers the question: *"What was each feature's contribution to this specific prediction, relative to a baseline?"* with mathematical fairness. However, SHAP does not capture **counterfactuals** or **causal reasoning**—that is the domain of the next family of interpretability methods.

3.3 Counterfactual and Contrastive Explanations

While feature attribution methods such as LIME and SHAP explain *why* a model made a particular prediction, **counterfactual explanations** address a complementary question:

> *"What minimal change to the input would have resulted in a different prediction?"*

Counterfactuals are inherently **contrastive**—they focus on what is *lacking* or should be *different* in order to achieve a desired outcome. This aligns closely with human reasoning: "If X had been the case instead of Y, the outcome would have changed."

Wachter, Mittelstadt, and Russell [55] proposed a seminal formulation of counterfactuals in machine learning via an optimization-based approach. Given an instance x such that $f(x) = y$, the goal is to find a nearby x^* such that:

$$f(x^*) = y_{\text{desired}} \quad \text{and} \quad x^* \approx x \tag{3}$$

This is typically achieved by solving:

$$\min_{x^*} \lambda \cdot \|x^* - x\| + \text{Loss}(f(x^*), y_{\text{desired}}) \tag{4}$$

where the first term encourages proximity to the original input, and the second enforces the desired prediction output.

Advantages. Counterfactual explanations are widely praised for their intuitive appeal. They resemble real data points and explicitly highlight the smallest changes needed to alter the model's decision. In practical applications like credit scoring, they offer actionable suggestions—for example, increasing income or reducing debt to obtain a loan.

Moreover, counterfactuals are **recourse-oriented**: they go beyond explanation to suggest feasible, user-understandable interventions that may flip a prediction to the desired class.

Challenges and Research Directions. Despite their promise, counterfactual methods face several key challenges:

- **Plausibility and feasibility.** Not all counterfactuals are realistic. Changing immutable attributes like age or race is unethical or meaningless. To address this, modern approaches introduce constraints on feature mutability and encourage generated instances to lie on the data manifold, enhancing realism.
- **Diversity of explanations.** Often, multiple valid paths lead to the desired output. Showing only one may mislead users. Methods like **DiCE** (Diverse Counterfactual Explanations) promote diversity using determinantal point processes, providing several distinct and plausible alternatives [35].
- **Model-agnosticism and computability.** Many counterfactual generation methods treat the model as a black box and rely only on querying outputs. Optimization techniques vary: gradient descent for differentiable models, heuristic search for discrete domains, and mixed-integer programming for rule-based or tree models.
- **Beyond binary classification.** Most early work focused on binary outputs. Recent methods extend counterfactuals to multiclass classification and regression tasks, but defining interpretable counterfactuals in these settings remains a challenging open problem.

Contrastive Reasoning. Building upon counterfactuals, Dhurandhar et al. [14] introduced the concepts of **pertinent positives** and **pertinent negatives**. A *pertinent positive* refers to the minimal subset of features sufficient to retain the original prediction—i.e., the essential components that justify the decision. In contrast, a *pertinent negative* refers to a feature whose absence is critical: had it been present, the prediction would have changed.

These ideas are typically operationalized through optimization procedures, often incorporating **autoencoder-based constraints** to ensure that the generated instances remain realistic. A canonical example is classifying an image as the digit "7" specifically because it lacks the loop that would have made it a "9."

Follow-up work by Luss et al. [34] extended this framework by introducing **logical constraints**—such as monotonicity and feature ordering—to better align explanations with human reasoning patterns.

Limitations. Despite their intuitive appeal, counterfactual explanations are not a panacea. A single counterfactual does not convey the *relative importance* of features, nor does it reflect the model's *global decision behavior*. Furthermore, if the model has learned spurious correlations from the training data, counterfactuals may inadvertently exploit these, resulting in suggestions to change irrelevant features.

Perhaps most importantly, not all proposed changes are **actionable**. Some may be infeasible, unethical, or beyond the user's control—such as suggesting someone "be older" or "change ethnicity"—thus limiting their real-world utility.

Summary. Counterfactual reasoning enhances interpretability by exploring *alternative realities* rather than merely explaining the present one. It complements attribution-based methods: for example, SHAP might reveal that a loan was denied due to low income, while a counterfactual explanation may show that increasing income by $5,000 could change the decision. Together, these methods provide a richer and more actionable understanding of model behavior.

Recent research emphasizes generating counterfactuals that are:

- **Valid:** Flip the decision;
- **Feasible:** Stay realistic;
- **Diverse:** Offer multiple alternatives;
- **Ethical:** Avoid suggesting changes in protected attributes.

These instance-specific, contrastive explanations provide a powerful tool for human-centered AI.

3.4 Example-Based Explanations and Prototypes

Another important class of post-hoc explanation techniques focuses on providing **examples** to help users understand model behavior. Instead of relying solely on feature attributions or symbolic rules, these methods retrieve or construct data points that serve as analogies, *prototypes*, or *criticisms* of the model's decisions.

This approach draws from the cognitive framework of **case-based reasoning**, wherein humans tend to understand and justify decisions by referring to concrete, representative examples—often contrasting "what is" with "what could have been."

Prototype Methods. One straightforward method is to retrieve training instances that are most similar to the input of interest—either in the raw input space or in a learned embedding space. For example, a nearest neighbor may serve as a basic explanation:

> *"This tumor was classified as malignant because it resembles these past malignant cases."*

However, naive nearest-neighbor approaches can be misleading if the chosen distance metric does not reflect the model's actual reasoning process.

To address this, more sophisticated techniques—such as **prototype learning networks**—incorporate example-based reasoning directly into the model architecture. One notable example is the **Prototypical Part Network (ProtoPNet)** proposed by Chen et al. [12], which learns a set of class-specific *prototypical image parts* during training. At inference time, the model compares patches from the input image to these learned prototypes. The final prediction is then based on the strength of similarity across these localized comparisons.

This produces intuitive, "*this-looks-like-that*" explanations. For instance:

> *"The bird is classified as a sparrow because this part of the input resembles the head of a prototypical sparrow."*

Because the prototypes are tied to actual training instances or components thereof, they can be visualized directly—offering compelling, human-aligned evidence for the model's decision.

Prototype-based models are typically trained with objectives that balance **predictive accuracy** and **interpretability**. They are encouraged to use a small number of prototypes to keep explanations simple, and to ensure that each prototype stays close to the training data distribution for realism. These constraints promote both transparency and fidelity to the underlying data.

Influence Functions and Training Attribution. Another prominent approach to explaining model behavior involves tracing predictions back to their most influential training examples using **influence functions** [26]. This method estimates how the model's output would change if a particular training point were perturbed or removed. In doing so, it provides attribution not to input features but to the training data itself.

For example, if a model misclassifies a test image of a cat, influence functions might trace this error back to a training image of a particularly odd-looking cat that closely resembles the input. In such cases, the explanation exposes the model's inductive biases and may help identify issues like overfitting or mislabeled data. Influence functions are especially useful for debugging and auditing models by highlighting problematic or overrepresented training points.

Criticisms and Outliers. While prototypes aim to represent typical examples within the data distribution, their natural counterpart—**criticisms**—highlight atypical or poorly represented instances. Kim et al. [24] proposed an approach that selects both prototypes and criticisms to offer a balanced, two-sided explanation.

Prototypes are often selected via clustering methods such as k-medoids and represent the core modes of the data distribution. Criticisms, in contrast, are data points that are *not well reconstructed* by the prototypes—typically outliers, edge cases, or examples from underrepresented regions of the input space. These help surface blind spots or biases in the model's understanding.

The joint use of prototypes and criticisms yields a richer explanatory narrative:

> *"Most cats look like these (prototypes), but these are unusual and might confuse the model (criticisms)."*

Rule Extraction as Examples. In some cases, model behavior is better conveyed not by individual examples but by abstracting to symbolic rules. Though not examples in the strict sense, **rule extraction techniques** like **TREPAN** offer interpretable surrogates for complex models by converting them into decision trees or rule lists.

TREPAN treats a trained neural network as a black-box oracle and incrementally grows a decision tree by querying it [13]. More recent approaches, such as **Bayesian rule lists** and other logical rule extraction frameworks, extend this idea to broader model families, including deep neural networks. When the fidelity between the rule-based surrogate and the original model is high, such explanations can strike a desirable balance between interpretability and faithfulness.

Applications in Images and Text. Example-based explanations have found particularly strong applications in **non-tabular domains**. In computer vision, while feature attribution methods such as saliency maps highlight individual pixels or regions, prototype-based methods offer a more holistic explanation by pointing to *entire regions* or even *whole training examples* that resemble the input image. This often provides a clearer, more human-comprehensible narrative.

In natural language processing, similar benefits emerge. Rather than interpreting token-level weights or attention scores—which may be too granular or noisy—retrieving similar sentences, phrases, or documents from the training corpus offers a more intuitive and semantically aligned explanation of the model's behavior.

Limitations. Despite their strengths, example-based explanations come with important caveats. When a test case is highly unique or anomalous, it may not resemble any prototype in the dataset, reducing the informativeness of the explanation. Additionally, algorithmically selected prototypes may not always align with human expectations or domain-specific notions of similarity.

To address these issues, some methods incorporate **training-time constraints** that enforce proximity between prototypes and real examples, while others leverage **user feedback** to guide prototype selection or pruning during deployment.

Summary. Example-based explanations embrace the principle of **"show, don't tell"**. Rather than quantifying feature importance or expressing symbolic rules, they offer concrete, case-based insights—often via analogy to similar past instances or canonical exemplars.

When integrated into model architectures (e.g., via prototype networks), such explanations are not merely post-hoc add-ons, but built-in components of the model's reasoning process. In domains like vision and NLP—where human understanding is often grounded in examples—these methods serve as valuable complements to feature attribution and rule-based techniques.

4 Visualization-Based Techniques (Saliency Maps and Beyond)

Modern machine learning—particularly deep learning—often operates on high-dimensional inputs such as images, text, or structured data with complex dependencies. **Visualization-based interpretability** techniques aim to make these complexities more visible and human-comprehensible.

The most well-known approach is the *saliency map* (or heatmap), but the field has since evolved into a rich ecosystem of visualization methods. In this section, we cover gradient-based saliency, class activation mapping (e.g., Grad-CAM), and other attribution visualizations, along with their theoretical underpinnings and limitations.

4.1 Gradient-Based Saliency and Attribution in Deep Networks

Saliency maps were among the earliest methods used to interpret convolutional neural networks (CNNs) [48]. The core idea is to compute the **gradient of the model output with respect to the input**:

$$\nabla_x f_{\text{class}}(x),$$

which indicates how sensitive the class-specific output is to each input pixel. The magnitude of the gradient at each pixel is visualized as a heatmap, highlighting regions that most influence the prediction.

Challenges. Raw gradients are often noisy and can be zero due to issues like *saturated neurons*, which leads to missing important features. Several methods have been proposed to address this:

- **Guided Backpropagation** [49]: Modifies the gradient flow by zeroing out negative gradients during backpropagation. This produces sharper and more focused saliency maps.
- **Integrated Gradients** [51]: Accumulates gradients along a straight-line path from a baseline input x' to the actual input x, computing:

  $$\text{Attribution}_i = (x_i - x'_i) \cdot \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i}\, d\alpha.$$

  This approach satisfies desirable axioms such as **completeness** (attributions sum to $f(x) - f(x')$) and **sensitivity**, and handles nonlinearity more robustly than raw gradients.
- **DeepLIFT** [46]: Compares neuron activations to a reference and propagates contributions via differences instead of derivatives. DeepLIFT has theoretical connections to Shapley values and served as a foundation for DeepSHAP.

These gradient-based techniques are widely appreciated for several reasons: they are highly efficient (requiring only one or a few forward/backward passes), model-specific (leveraging internal gradients), and visually interpretable. In computer vision, especially, saliency maps and heatmaps offer intuitive insights into what parts of an image the model “attends to” when making decisions.

Limitations. Despite their popularity, gradient-based visualization methods have several significant limitations. One major concern is their **inconsistency**: different techniques (e.g., vanilla gradients, Guided Backprop, Integrated Gradients) often produce visually distinct saliency maps for the same input and model, raising questions about their reliability and interpretive value.

Moreover, these visualizations may not faithfully reflect the model’s true decision-making process. They can sometimes highlight irrelevant **low-level input features**, structural artifacts, or texture priors—factors that are not causally responsible for the prediction.

A particularly influential critique was offered by Adebayo et al. [2], who introduced a suite of *sanity checks* for saliency methods. In a striking finding, they showed that some saliency maps remain largely unchanged even after **randomizing the model weights**—

implying that these explanations were independent of the model's learned parameters and thus untrustworthy.

These observations have led to a push for more rigorous evaluation protocols in the interpretability community. Suggested validation approaches include:

- **Randomization tests:** Randomizing model weights to assess dependency on learned parameters;
- **Model ablation:** Removing or altering components of the network and observing changes in explanations;
- **Perturbation metrics:** Systematically modifying input features to see if the explanation corresponds to actual prediction shifts.

In summary, gradient-based visualization techniques offer intuitive and computationally efficient insights into deep model behavior, especially in vision and text domains. However, they must be interpreted with caution, ideally in conjunction with statistical validation or theoretical guarantees to ensure explanatory *faithfulness*.

4.2 Class Activation Maps and Grad-CAM

While vanilla saliency maps and guided backpropagation highlight individual input features that affect a model's score, **Class Activation Mapping (CAM)** techniques aim to localize entire *objects or semantic concepts* in the input space.

The original CAM method by Zhou et al. [58] introduced a specific architectural requirement: the network must terminate with **Global Average Pooling (GAP)** followed by a fully connected layer. For a target class c, the final score is computed as a weighted sum of the spatially pooled feature maps. These same weights are then used to produce a class-specific activation map:

$$L^c = \sum_k w_k^c A^k,$$

where w_k^c is the learned weight connecting feature map A^k to class c. The activation map L^c is then upsampled to match the input resolution, yielding a heatmap that highlights spatial regions most responsible for the class prediction.

CAM's appeal lies in its ability to produce intuitive, object-level visual explanations, but its architectural constraints limit its applicability to a narrow class of models—prompting the development of more general methods like Grad-CAM.

Grad-CAM. **Grad-CAM** (Gradient-weighted Class Activation Mapping) [43] generalizes CAM to a broader range of CNN architectures without requiring architectural changes.

Instead of using weights from a fully connected layer, Grad-CAM computes the **gradients of the class score** y^c with respect to the feature maps A^k in the final convolutional layer:

$$\alpha_k^c = \frac{1}{Z} \sum_{i,j} \frac{\partial y^c}{\partial A_{ij}^k},$$

where Z is the number of spatial positions in feature map A^k. These α_k^c values represent the importance of each feature map k for the target class c. The final class activation map is computed as:

$$L_{ij}^c = \text{ReLU}\left(\sum_k \alpha_k^c A_{ij}^k \right),$$

where the ReLU ensures only features that have a positive influence on the class score are retained.

The result is a coarse localization heatmap that highlights the regions in the input image most influential for the prediction. For example, in an image classified as "fire truck," Grad-CAM might highlight the fire truck's region while leaving the background dim.

Guided Grad-CAM. While Grad-CAM provides class-discriminative localization, it lacks pixel-level precision. On the other hand, **Guided Backpropagation** offers high-resolution details but lacks class specificity. **Guided Grad-CAM** combines the strengths of both: it multiplies the Grad-CAM heatmap with Guided Backpropagation gradients element-wise, producing visually sharp and semantically focused saliency maps.

Beyond Images: Text and Structured Data. Although originally developed for vision, visualization-based interpretability methods are also applied to other data modalities:

- In **text classification**, saliency scores can be computed over input tokens using gradient-based methods or attention distributions.
- In **structured/tabular data**, *feature-importance heatmaps*, *partial dependence plots*, and *accumulated local effects* plots can visualize how predictions vary with input features.
- In **transformer models**, **attention maps** naturally lend themselves to visualization. These can show which input tokens influence a prediction—for instance, which words a QA model attends to when answering a question.

Caveats: Is Attention Explanation? Despite the intuitive appeal of attention maps, their status as faithful explanations is debated. Serrano and Smith [44] and Jain and Wallace [22] demonstrated that modifying attention weights often does not change the model's output, suggesting that attention may not always reflect *causal importance*. As a result, attention should be treated as a visualization aid rather than a definitive explanation method unless validated through perturbation or causal analysis.

Summary. CAM and Grad-CAM remain cornerstone tools for visualizing model decisions in CNNs. They are widely adopted in computer vision for producing semantically meaningful heatmaps and have inspired adaptations across data modalities. However, as with all interpretability techniques, their outputs must be interpreted with care—preferably alongside validation strategies such as perturbation tests, counterfactuals, or attribution consistency checks.

4.3 Other Visualization Methods

Beyond saliency and CAM-based approaches, a diverse set of **visualization-based techniques** has emerged to enhance interpretability in complex models.

Feature Visualization via Activation Maximization. Instead of explaining a specific input, one can explore what a **neuron or class has learned** by synthesizing an input that *maximally activates* it. This is done through solving the following optimization:

$$x^* = \arg\max_x f_{\text{neuron}}(x) - \lambda R(x),$$

where $f_{\text{neuron}}(x)$ is the activation of the target neuron (or class logit), and $R(x)$ is a regularization term to ensure naturalness and prevent noise artifacts. This technique—popularized in works by Erhan et al. [16] and later Olah et al. [36]—produces **synthetic images** that represent "ideal stimuli" for specific neurons. For example, a neuron associated with "flamingo" might produce pink textures and leg-like shapes.

Modern variants enhance realism through the use of pretrained generator networks, texture priors, and image priors to produce high-fidelity and semantically meaningful visuals.

Projection and Embedding Visualization. In many deep models, particularly autoencoders, language models, and vision transformers, data is embedded into **high-dimensional latent spaces**. Visualization of these learned representations can be achieved by applying dimensionality reduction techniques such as **t-SNE** or **UMAP**.

Projecting embeddings to 2D or 3D enables exploration of the **semantic geometry** of the latent space. For instance, in image classification, different dog breeds may cluster in the embedding space, suggesting that the model has learned to separate them meaningfully. These projections are often used for sanity checks, anomaly detection, and qualitative evaluations of representation learning.

Partial Dependence and ICE Plots. While less frequently applied to deep learning, **Partial Dependence Plots (PDPs)** and **Individual Conditional Expectation (ICE)** plots remain essential tools for **visualizing feature effects**, especially in tabular models.

- **PDPs** show the *average model prediction* as a function of a selected input feature, marginalizing over other features. This provides a **global view** of how a single feature influences the output.
- **ICE plots** go further by displaying one line per instance, revealing **heterogeneous feature effects** across the dataset.

However, both PDPs and ICE can be misleading when input features are correlated. To address this, **Accumulated Local Effects (ALE)** plots [5] have been proposed. ALE plots compute *local effects* by averaging derivative-based differences in small intervals, thus avoiding extrapolation and reducing bias caused by feature correlations.

These methods remain particularly valuable in domains where interpretability of individual features is paramount—such as healthcare, finance, and scientific applications.

Strengths and Limitations. Visualization-based methods offer several compelling advantages. They provide immediate and intuitive insight into a model's decision-making process by visually highlighting the regions or features most responsible for a prediction. Their modularity and adaptability allow them to be applied across architectures—including convolutional neural networks, transformers, and even gradient-boosted trees—and across modalities such as vision, language, and structured data.

However, these methods also face important limitations. For example, saliency maps often produce *broad or ambiguous heatmaps*, making it difficult to pinpoint the exact features driving the prediction. Moreover, they typically show **where** the model is focusing, but not necessarily **how** those features influence the decision. Another challenge is evaluation: a common validation technique involves *removal tests*, where salient regions are occluded and the resulting drop in model confidence is used as a proxy for explanation fidelity. Yet, these tests are indirect and not always reliable.

Emerging Directions. Several recent efforts aim to overcome these limitations by providing more **semantically meaningful and causally grounded visual explanations**. One direction involves **counterfactual visual explanations**, which aim to identify the minimal change in an image that would alter the model's prediction. This yields direct insight into decision boundaries and actionable feedback.

Another innovative approach is **Concept-Based Visual Explanations**, exemplified by the **TCAV (Testing with Concept Activation Vectors)** framework [25]. Instead of attributing importance to individual pixels or patches, TCAV measures how strongly a human-defined concept (e.g., “stripes”, “wheels”) contributes to a model’s output. It does so by computing directional derivatives in the activation space defined by concept vectors. This bridges the gap between raw model features and human-understandable abstractions, enabling more cognitively aligned explanations.

Conclusion. Visualization techniques have become indispensable tools for understanding and debugging deep learning models. They help practitioners detect model flaws—for example, when attention is misdirected toward irrelevant background patterns—and foster greater trust by aligning machine decisions with human reasoning.

While no single visualization method is universally optimal, the complementary use of multiple techniques—saliency, attribution, concept-based reasoning, counterfactuals—forms a **robust and versatile interpretability toolkit**. As AI systems continue to grow in complexity and impact, such interpretability tools will be essential for ensuring transparency, accountability, and human trust.

5 Hybrid and Emerging Interpretability Approaches

As the field of explainable AI (XAI) matures, researchers have begun to explore **hybrid approaches** that embed interpretability directly into the model architecture, training objective, or constraints. These methods blur the line between *intrinsic* and *post-hoc* interpretability by encouraging explanations to emerge naturally during inference—rather than being retrofitted afterward.

5.1 Concept-Based and Self-Explaining Models

One promising direction is to build models that reason via high-level, human-understandable **concepts**.

Concept Bottleneck Models (CBM). **CBMs** [27] decompose prediction into two stages: an encoder maps the input x to a vector of concept predictions $c(x)$, followed by a classifier $f(c(x))$. The model is trained with supervision on both the concepts and final outputs. This architecture enables:

- **Inspection:** The intermediate concept vector $c(x)$ can be interpreted directly, revealing which concepts were active.
- **Intervention:** Users can modify concept values to observe how changes affect the output prediction.

For example: *"Zebra predicted because concepts 'stripes' and 'four-legged' were detected."* CBMs thus support both transparency and user interaction.

Testing with Concept Activation Vectors (TCAV). **TCAV** [25] is a post-hoc method that quantifies the influence of human-defined concepts on predictions. Given a set of examples that represent a concept C, TCAV computes a **concept vector** in the model's latent space. The directional derivative along this vector is then used to evaluate:

$$\text{TCAV Score} = \mathbb{E}_x \left[\frac{\partial f(x)}{\partial v_C} \right],$$

indicating the sensitivity of the model's output to concept C. TCAV has been used to identify *spurious correlations* (e.g., watermark presence) and to validate that the model's reasoning aligns with expected factors (e.g., "smiling" influencing "happy").

Self-Explaining Neural Networks (SENN). **SENN** [3] introduces an architecture that produces both predictions and explanations in a single forward pass. The model outputs are given by:

$$f(x) = \sum_i \alpha_i(x) \cdot h_i(x),$$

where $h_i(x)$ are interpretable basis concepts, and $\alpha_i(x)$ are their associated relevance scores. SENN enforces regularization constraints during training—such as sparsity, disentanglement, and Lipschitz continuity—to improve the interpretability and stability of the learned explanations.

5.2 Prototype and Example-Based Networks

Prototype-based models build interpretability into the model by reasoning through comparisons with learned prototypes or training examples.

Prototype Networks (ProtoPNet). **ProtoPNet** [12] compares the latent representation of input $z(x)$ to a set of learned prototypes p_j, each of which is tied to a **visual patch from the training set**. The class-specific prediction score is computed via:

$$\text{score}_c(x) = \sum_{j \in \text{class } c} w_j \cdot \text{sim}(z(x), p_j),$$

where sim$(\cdot)$ denotes a similarity function (e.g., negative distance). This yields intuitive "*this-looks-like-that*" explanations. Each prototype can be visualized, inspected, and even *edited*, making ProtoPNet models suitable for transparent decision-making and model debugging.

5.3 Explanation Regularization

Some hybrid interpretability approaches take a different route by introducing **explanation regularization**—training-time constraints that explicitly penalize *uninterpretable* behavior. Rather than relying solely on post-hoc analysis, these techniques integrate interpretability into the model's loss function.

One seminal work in this direction is by Ross et al. [40], who proposed a gradient-based regularization scheme. In their method, a model's saliency map—i.e., the gradient of its prediction with respect to the input—is encouraged to align with a **user-specified explanation mask**. This alignment is enforced through a regularization term added to the loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \cdot \left\| \nabla_x f(x) \odot (1 - M) \right\|^2 ,$$

where M is the binary mask indicating which input features are deemed relevant by the user. The goal is to minimize gradient magnitude on irrelevant features—thus forcing the model to focus its attention where humans expect it to.

Beyond gradient alignment, other forms of explanation regularization include:

- **Sparsity constraints:** Encouraging explanations (e.g., attribution vectors) to be sparse for easier human interpretation.
- **Concept supervision:** Penalizing deviations from human-defined concepts or concept-based reasoning paths.
- **Disentanglement regularization:** Promoting orthogonal or independent latent features for clearer interpretation.

These strategies allow models to develop internal representations that are both *predictively useful* and *semantically aligned* with human expectations.

5.4 Strengths and Open Challenges

Explanation-regularized models offer several notable strengths. Most significantly, the explanations they provide are **faithful by construction**, as they emerge directly from the training objective. This removes the need for separate post-hoc explanation procedures and ensures consistency between model behavior and its interpretation.

These models also support **human-in-the-loop learning**. By incorporating human-labeled masks or concept annotations, users can intervene during training to steer the model's attention or decision-making logic—enabling greater transparency and control.

However, several challenges remain. One major issue is the need for **additional supervision**: saliency masks, concept labels, or annotations must be defined, which can be costly or infeasible in many domains. Even when such supervision is available, learned internal representations may not align perfectly with intended semantics. For example, a concept like "cloudiness" might inadvertently capture image brightness or texture patterns.

Furthermore, designing architectures and loss functions that balance **interpretability and performance** is an ongoing area of research. These models often face trade-offs in accuracy, generalization, or training complexity. The right balance depends heavily on the domain, data, and the nature of the task.

Summary. Hybrid approaches signal a broader shift toward **explainability-by-design**. Rather than treating interpretability as an afterthought, these methods integrate transparency directly into the training process. Whether through concept supervision, prototype-based structures, or learned explanation weights, they aim to unify accuracy and interpretability in a single framework. This convergence makes them promising candidates for deployment in high-stakes, real-world applications, where understanding the *why* behind a prediction is just as important as the prediction itself.

5.5 Interactive and Composite Systems

Interpretability can also emerge from **interactive systems** or through the combination of multiple model components in a modular architecture. These approaches move beyond static, one-shot explanations and enable richer, user-driven interpretability workflows.

Two-Stage Models. In natural language processing and computer vision, **two-stage architectures** have been proposed where one sub-model generates an **explanation or rationale**, and another sub-model makes the final prediction—potentially conditioned on that explanation. For example, the first stage might produce:

"The review is classified as positive because it praises the actor's performance."

The second stage then verifies or utilizes this rationale to ensure it faithfully supports the decision. These systems often require training on datasets with human-annotated explanations (e.g., e-SNLI, CoS-E), and additional constraints are applied to avoid generating plausible-but-unfaithful justifications. A common technique is to enforce **rationale grounding**, where the prediction model must attend to the same input tokens identified in the rationale [57, 10].

Interactive Tools. Beyond model-level innovations, a growing body of work focuses on **interactive interpretability tools** that allow users to explore, manipulate, and interrogate models in real-time. These tools typically support:

- **Instance-level analysis:** Inspecting explanations for individual predictions.
- **Global behavior analysis:** Aggregating explanations across data to identify trends or biases.
- **Perturbation-based exploration:** Modifying input features to examine counterfactual changes in prediction.

Tools such as `LIT` (Language Interpretability Tool) [52], `AllenNLP Interpret` [56], and `TruLens` provide interfaces for model introspection, often with visualization support (e.g., saliency overlays, attention maps, concept activations).

To support dynamic exploration, underlying algorithms must generate explanations with **low latency and high responsiveness**. As such, lightweight surrogate models, precomputed embeddings, and optimized counterfactual search methods are often employed. These systems not only improve usability but also empower domain experts to engage with models in a *diagnostic, hypothesis-driven* fashion.

5.6 Summary: Hybrid Approaches

Hybrid interpretability methods represent a significant step toward bridging the gap between model reasoning and **human-understandable concepts and structures**. Unlike traditional black-box models that output opaque predictions, hybrid models aim to reflect decision-making processes that align with domain-specific knowledge.

For example, a diagnostic model that reasons over symptom-level concepts and mimics the diagnostic logic of a clinician is more likely to be trusted by users. Similarly, a sentiment classifier that justifies its decisions through coherent natural language rationales fosters user comprehension and transparency. By embedding interpretability into the architecture or learning objective, these methods enable models to explain themselves in ways that are intuitive and aligned with human cognition.

Limitations. Despite their appeal, hybrid models face several challenges. In many real-world tasks, it remains unclear what constitutes a valid set of high-level, interpretable concepts—if such concepts exist at all. Even when they do, annotating them often demands significant human expertise and can become prohibitively expensive at scale. Additionally, the very constraints that make models more interpretable can also restrict their expressive power, leading to a potential degradation in predictive performance.

A particularly concerning issue is the phenomenon of **bypassing**, where the model learns to ignore the interpretable components (e.g., concept layers or prototypes) and instead relies on spurious or uninterpretable features to make predictions. In such cases, explanations may appear plausible while being disconnected from the model's true decision process—undermining the goal of transparency.

To address these limitations, recent work focuses on improving the alignment between internal representations and human semantics. Techniques like *disentangled representation learning* seek to ensure that latent features correspond to meaningful and independent concepts. *Human-in-the-loop* learning provides mechanisms for users to iteratively correct or refine explanations. Furthermore, a growing body of research is developing quantitative metrics for evaluating how faithfully a model's internal reasoning aligns with human-understandable logic, providing more rigorous tools for assessing interpretability.

Conclusion. Hybrid interpretability approaches embody a compelling vision: building **models that explain themselves by design**. Rather than bolting explanations onto an opaque system post-hoc, this paradigm embeds transparency into the architecture itself. The resulting systems not only deliver accurate predictions, but also offer structured, auditable rationales for their behavior.

This shift—from "black-box plus explanations" to "transparent-by-construction"—marks a foundational evolution in interpretable AI. As AI systems increasingly influence high-stakes domains such as healthcare, finance, and law, the ability to produce faithful, accessible, and trustworthy explanations will be critical for ensuring fairness, accountability, and societal trust.

6 Discussion: Strengths, Limitations, and Comparative Analysis

In this section, we synthesize the interpretability methods discussed throughout the review by comparing them along key dimensions: **intrinsic vs. post-hoc**, **local vs. global**, **model-agnostic vs. model-specific**, and their **fidelity, stability**, and **human interpretability**. These axes provide a framework for understanding the trade-offs inherent in different explanation paradigms.

6.1 Intrinsic vs. Post-hoc

Intrinsic interpretability is achieved by design. In models like sparse linear models, decision trees, or generalized additive models, the internal structure itself serves as the explanation. There is no need to approximate the model with external surrogates. The main advantage lies in faithfulness: the model's logic is transparent and inspectable.

However, such models may struggle to match the accuracy of deep or ensemble-based architectures, especially in high-dimensional or complex domains.

By contrast, **post-hoc methods** generate explanations *after* the model is trained, typically for black-box systems. These include methods like LIME, SHAP, counterfactuals, or saliency maps. Their flexibility allows interpretability without sacrificing model performance. The central concern here is **fidelity**—do these explanations accurately reflect the model's true reasoning? Theoretical analyses (e.g., Garreau & von Luxburg, 2020) have shown that local approximations may misrepresent global logic or feature interactions.

6.2 Local vs. Global Explanations

Local methods provide explanations for individual predictions. LIME, SHAP, counterfactuals, and influence functions all aim to answer questions like: *"Why did the model make this decision for this input?"* These are useful in domains where specific decisions require justification or recourse.

Global methods, on the other hand, characterize the overall behavior of the model. Examples include feature importance rankings, partial dependence plots, and interpretable surrogates like decision trees. These help in understanding general trends, bias detection, or model auditing.

In practice, combining both perspectives is highly effective: local explanations assist end-users or practitioners, while global views serve developers and regulators.

6.3 Model-Agnostic vs. Model-Specific

Model-agnostic techniques—such as Kernel SHAP or LIME—treat the model as a black-box, querying it without relying on internal gradients or architectures. This makes them widely applicable across model types but potentially less efficient or accurate.

Model-specific methods—like Tree SHAP or Grad-CAM—leverage internal structures for optimized explanation. They can often provide exact or more stable results, but are tied to particular architectures (e.g., trees or CNNs). Many practical toolkits now hybridize these methods, selecting the most efficient explainer depending on model type.

6.4 Explanation Formats

Explanations can take many forms, each suited to different users and tasks:

- *Feature attributions* (e.g., SHAP, LIME) assign importance scores or visual highlights. - *Counterfactuals* suggest minimal changes needed to alter a prediction. - *Rules and decision lists* offer logical, symbolic rationales. - *Examples and prototypes* point to similar training cases. - *Concept-based* explanations reason using human-aligned semantics (e.g., TCAV, CBM).

Choosing the right format is context-dependent. A clinician may prefer example-based reasoning (*"this tumor looks like that one"*), while legal or regulatory domains may demand rule-based or symbolic justifications.

Overall, each method brings unique strengths and trade-offs. In critical applications, a combination of complementary approaches often provides the best coverage—balancing faithfulness, interpretability, and usability.

6.5 Strengths and Limitations Summary

Method	**Strengths**	**Limitations**
Linear models	Transparent, coefficients directly interpretable	Limited expressiveness for nonlinear patterns
Decision trees	Intuitive if shallow; path-based explanations	Fragile when deep; prone to overfitting
LIME	Model-agnostic; locally faithful; intuitive output	Sensitive to sampling; unstable across runs; approximative
SHAP	Theoretically grounded; consistent; additive explanations	Computationally intensive; independence assumptions
Saliency / Grad-CAM	Visual and efficient for CNNs; pixel-level heatmaps	Can be noisy; interpretation not always semantically aligned
Counterfactuals	Actionable; user-relevant; recourse-oriented	Feasibility and causality concerns; not all changes are realistic
Prototypes / examples	Human-aligned reasoning; case-based explanations	Dependent on similarity metrics; vulnerable to low-quality data

Table 1: Comparative overview of popular interpretability methods

6.5.1 Stability and Robustness

Explanation robustness remains a key concern. While SHAP (especially Tree SHAP) is deterministic and theoretically stable, methods like LIME introduce stochasticity due to sampling, which can lead to varying explanations across runs. Furthermore, studies (e.g., Ghorbani et al., 2019) have shown that some attribution methods are susceptible to adversarial perturbations that alter explanations without affecting predictions. Robust interpretability demands both algorithmic regularization and diagnostic checks—such as sanity tests, ablation studies, and input perturbation analyses.

6.5.2 Evaluation Criteria

Evaluating interpretability is nontrivial. Several dimensions have emerged as key:

- **Fidelity:** Does the explanation accurately reflect the model's reasoning?
- **Simplicity:** Is the explanation cognitively manageable and parsimonious?
- **Usefulness:** Does it support decision-making, debugging, or trust calibration?
- **Comprehensiveness:** Does it capture the full scope of influential factors?

Increasingly, **user studies** are employed to assess usefulness and trust in human-AI interaction settings, complementing algorithmic benchmarks.

6.6 Complementarity of Methods

Interpretability methods are not mutually exclusive—they are best viewed as **complementary lenses** into model behavior. SHAP offers both local and global feature attribution with theoretical guarantees. LIME and Anchors produce sparse, rule-based rationales that appeal to human logic. Counterfactuals inject a causal and actionable dimension, answering "what could have been done differently?" Prototypes and example-based methods allow case-wise analogies, while concept-based approaches (e.g., TCAV, CBMs) elevate reasoning to human-aligned abstractions.

In visual domains, saliency and Grad-CAM continue to serve as essential tools for spatially grounding model attention. Importantly, when different methods produce conflicting explanations, this divergence can act as a diagnostic signal—flagging regions of model nonlinearity, ambiguity, or insufficient representation. Such disagreements are not necessarily failures but opportunities for deeper auditing and trust calibration.

6.7 Recent Trends and Perspectives

The field of explainable AI (XAI) continues to evolve rapidly, driven by both theoretical inquiry and practical demand. Several emerging trends are shaping the next generation of interpretability research.

On the theoretical front, there is growing attention to the **axiomatic foundations** of interpretability methods. Researchers are increasingly formalizing the trade-offs between desirable properties such as *completeness*, *consistency*, and *stability*, with some impossibility results suggesting that not all can be simultaneously satisfied. This theoretical framing has deepened our understanding of the limitations and potential of different explanation families.

At the application level, interpretability is becoming increasingly **domain-specialized**. Techniques are now being tailored for specific data modalities—including text, images, graphs, and time series—each with distinct structures and constraints. For example, interpretable methods for graphs (e.g., GNNExplainer) must respect relational topology, while NLP methods contend with token-level ambiguity and sequential dependencies.

Another major development is the incorporation of **causal reasoning** into interpretability. Moving beyond correlation, causal interpretability seeks to uncover underlying data-generating mechanisms and provide counterfactual insights grounded in causal models. This shift aligns with broader efforts in trustworthy AI to ensure explanations support reasoning about *why* outcomes occur and *what would have happened otherwise*.

Interpretability is also increasingly employed in service of **fairness and accountability**. XAI tools are now used to audit models for biased behavior across demographic subgroups, identify disparate impact, and assess procedural fairness. This reflects a shift toward **human-centered evaluation**, where the effectiveness of an explanation is judged not solely by technical metrics but by how well it fosters user *understanding*, *trust*, and *actionability*.

Finally, it is increasingly recognized that different stakeholders—developers, end-users, regulators—have different goals, capacities, and constraints. As such, the choice of interpretability method should be informed by the **intended use case, user context**, and **model architecture**. A single technique rarely suffices. Instead, triangulating multiple complementary explanations—combining attribution, example-based reasoning, counterfactuals, and conceptual insights—provides a richer and more trustworthy interpretability toolkit.

7 Conclusion

The field of interpretable AI has made significant strides in the past five years, evolving from a handful of heuristic approaches into a rich collection of theoretically grounded and practically evaluated techniques. This review highlighted both foundational methods and recent innovations across the interpretability spectrum, with particular attention to **LIME** and **SHAP**—exemplars of post-hoc explanations balancing flexibility and theoretical guarantees. We dissected their mathematical formulations and traced how they laid the groundwork for further developments.

From intrinsically interpretable models to post-hoc methods such as local surrogates, counterfactual explanations, visualization tools, and hybrid models, each method offers a different lens for understanding model behavior.

A recurring theme throughout this survey is that **no single method suffices for all interpretability needs**. Each class of interpretability methods contributes a distinct perspective on model behavior. For instance, **feature attribution** techniques such as LIME, SHAP, and gradient-based methods help answer the question of *which features* influenced a prediction and by *how much.* In contrast, **counterfactual and contrastive explanations** shed light on *what could have changed* to alter the outcome, offering actionable or causal interpretations.

Rule-based and surrogate models provide interpretable approximations of black-box models, typically using decision trees or logical rules to mimic behavior. **Example-based explanations**, by grounding predictions in real or prototypical data instances, enhance the relatability of the output. **Visualization techniques** remain essential, particularly in domains like computer vision, NLP, and structured data, where saliency maps or attention heatmaps can offer human-aligned insights. Finally, **concept-based and self-explaining models** embed interpretability directly into the architecture, allowing explanations to emerge naturally as part of the model's forward pass.

Throughout the discussion, we examined the trade-offs between **fidelity**, **simplicity**, **generality**, and **user interpretability**. It is evident that evaluating explanations should not be limited to technical soundness alone; their practical utility for human users is equally critical. Metrics such as stability, faithfulness, and cognitive usability are increasingly recognized as being just as important as predictive accuracy.

Looking forward, the future of interpretable AI lies in three converging directions:

- **Context-aware and user-centered explanations**, where the form and content of the explanation are adapted to the user's role, expertise, and decision needs.
- **Causality-enhanced interpretability**, which integrates causal models and counterfactual reasoning to move beyond correlation and uncover deeper mechanisms behind predictions.
- **Transparent-by-design systems**, where interpretability is not an afterthought but an integral part of model architecture, training objectives, and evaluation protocols.

In an era where AI systems increasingly influence high-stakes decisions in healthcare, law, finance, and public policy, interpretability is not a luxury—it is a prerequisite. Achieving trustworthy AI demands methods that are not only accurate but also **understandable, auditable, and aligned with human values**. As the field matures, the integration of interpretability with fairness, robustness, and accountability will define the next generation of AI systems—systems that explain not just what they predict, but why.

7.1 Future Directions

Several promising directions are shaping the next frontier of explainable AI. One persistent challenge is **scalability and efficiency**: as deep learning models grow in complexity and datasets become increasingly high-dimensional, explanation techniques must scale accordingly without incurring prohibitive computational overhead.

Closely related is the need for **robustness**. Explanations should remain stable under small input perturbations and be resilient to adversarial manipulation, ensuring that interpretability tools remain trustworthy in real-world deployments. Developing explanation methods with provable robustness guarantees is an ongoing research frontier.

On the theoretical front, there is increasing interest in the **unification** of diverse explanation paradigms. Bridging feature attribution, counterfactual reasoning, rule-based logic, and concept-based representations could lead to hybrid frameworks that synthesize the strengths of multiple approaches. Such unified architectures may enable more comprehensive and modular interpretability pipelines.

Another key direction involves embedding **causal reasoning** and integrating **domain knowledge** into the explanation process. Causally informed explanations go beyond surface-level correlations to uncover underlying mechanisms—an essential feature in safety-critical domains such as medicine, law, and autonomous systems.

As AI systems enter **multi-modal, dynamic, and interactive environments**—from reinforcement learning agents to multi-agent systems—novel interpretability techniques must evolve to capture temporal dependencies, long-term behavior, and emergent properties across modalities.

Perhaps most crucially, the field is increasingly shifting toward **human-centered evaluation**. Empirical user studies, cognitive modeling, and participatory design are needed to determine which explanations genuinely foster *understanding*, *trust*, and *safe decision-making* for diverse stakeholders. Moving from algorithm-centered benchmarks to people-centered impact assessments marks a major paradigm shift for the next phase of XAI.

In conclusion, interpretability has become a foundational pillar of responsible AI—on par with accuracy, efficiency, and fairness. Over the past five years, explainable AI has matured into a vibrant, interdisciplinary subfield, underpinned by rigorous theoretical principles and increasingly integrated into practical workflows.

As the field progresses, we foresee an ecosystem of **complementary interpretability techniques**, informed by game theory, optimization, information theory, and human-computer interaction, collectively working to advance transparency, accountability, and alignment with human values.

We envision a future where machine learning systems can *explain their decisions as naturally and clearly as they make them*—enabling more meaningful human-AI collaboration, fostering trust, and ensuring the safe, transparent deployment of intelligent technologies in society.

References

[1] Ahmed Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[3] David Alvarez-Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

[4] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 35–44, 2017.

[5] Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(4):1059–1086, 2020.

[6] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[8] Dimitris Bertsimas and Jack Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, 2017.

[9] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and Regression Trees.* CRC press, 1984.

[10] Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[11] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730, 2015.

[12] Chaofan Chen, Oscar Li, Alina Tao, Allison Barnett, Cynthia Rudin, and Jonathan Su. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[13] Mark W Craven and Jude W Shavlik. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems*, pages 24–30, 1996.

[14] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Pulin Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[15] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[16] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. In *University of Montreal Technical Report*, 2009.

[17] Fabrizio Esposito, Donato Malerba, and Giovanni Semeraro. Simplifying decision trees by pruning and expert collaboration. *Applied Artificial Intelligence*, 11(2):195–217, 1997.

[18] Damien Garreau and Ulrike von Luxburg. Explaining the explainer: A first theoretical analysis of lime. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, pages 1287–1296. PMLR, 2020.

[19] David Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), Program Information*, 2017.

[20] Maya Gupta, Andrew Cotter, Jan Pfeifer, Konstantin Voevodski, Kevin Canini, Andrei Mangylov, Wojciech Moczydlowski, and Andre Van Esbroeck. Monotonic calibrated interpolated look-up tables. In *Advances in Neural Information Processing Systems*, pages 2082–2090, 2016.

[21] Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical Science*, 1(3):297–318, 1986.

[22] Sarthak Jain and Byron C Wallace. Attention is not explanation. In *Proceedings of NAACL-HLT*, pages 3543–3556, 2019.

[23] Ritesh Khandelwal, Ankur Sinha, and Ankur Agrawal. Additive-feature-attribution methods: A review on explainable artificial intelligence for fluid dynamics and heat transfer. *arXiv preprint arXiv:2409.11992*, 2023.

[24] Been Kim, Rajiv Khanna, and Oluwasanmi Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[25] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 2668–2677, 2018.

[26] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1885–1894, 2017.

[27] Pang Wei Koh, Shiori Sagawa, Hamed Hassanzadeh Marklund, Sang Michael Xie, Fanny Yang Zhang, Percy Liang, and Tatsunori Hashimoto. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5338–5348, 2020.

[28] Himabindu Lakkaraju, Stephen Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684, 2016.

[29] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.

[30] Yinxi Li, Jian Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[31] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD In-*

ternational Conference on Knowledge Discovery and Data Mining, pages 623–631, 2013.

[32] Scott M Lundberg, Gabriel Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[33] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

[34] Ronny Luss, Amit Dhurandhar, Pin-Yu Chen, Karthikeyan Shanmugam, and Payel Das. Generative pertinent explanations with constraints (gp2). In *Workshop on Human Interpretability in Machine Learning (WHI)*, 2019.

[35] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT* 2020)*, pages 607–617, 2020.

[36] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017.

[37] J Ross Quinlan. Simplifying decision trees. In *International journal of man-machine studies*, volume 27, pages 221–234, 1987.

[38] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

[39] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[40] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2662–2670, 2017.

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

[42] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks

via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[43] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[44] Sofia Serrano and Noah A Smith. Is attention interpretable? In *Proceedings of ACL*, pages 2931–2951, 2019.

[45] Lloyd S Shapley. A value for n-person games. pages 307–317, 1953.

[46] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 3145–3153, 2017.

[47] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2014.

[48] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2014.

[49] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations (ICLR)*, 2015.

[50] Mukund Sundararajan and Amir Najmi. Many shapley values for model explanation. *arXiv preprint arXiv:2005.00623*, 2020.

[51] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328, 2017.

[52] Ian Tenney, James Wexler, Jasmijn Bastings, Ellie Pavlick Wang, Frank Bertsch, Adam Poliak, Abigail Rosenberg, Dipanjan Das, Sebastian Ruder, and Benjamin Van Durme. The language interpretability tool: Extensible, interactive visualizations and analysis for nlp models. In *Proceedings of EMNLP: System Demonstrations*, pages 107–118, 2020.

[53] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

[54] Berk Ustun and Cynthia Rudin. Optimized risk scores. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1125–1134, 2017.

[55] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31(2):841–887, 2017.

[56] Eric Wallace, Shi Feng Wang, Su Li, Sameer Singh, and Matt Gardner. Allennlp interpret: A framework for explaining predictions of nlp models. In *Proceedings of EMNLP-IJCNLP: System Demonstrations*, pages 7–12, 2019.

[57] Omar Zaidan and Jason Eisner. Using annotated text to learn rationale classifiers. In *Proceedings of NAACL-HLT*, pages 355–363, 2007.

[58] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016.