Toward Transparent AI: A Survey on Interpreting the Inner Structures of Deep Neural Networks

Tilman Räuker*
traeuker@gmail.com

Anson Ho*
Epoch
anson@epochai.org

Stephen Casper*
MIT CSAIL
scasper@mit.edu

Dylan Hadfield-Menell
MIT CSAIL

***Abstract*—The last decade of machine learning has seen drastic increases in scale and capabilities. Deep neural networks (DNNs) are increasingly being deployed in the real world. However, they are difficult to analyze, raising concerns about using them without a rigorous understanding of how they function. Effective tools for interpreting them will be important for building more trustworthy AI by helping to identify problems, fix bugs, and improve basic understanding. In particular, "inner" interpretability techniques, which focus on explaining the internal components of DNNs, are well-suited for developing a mechanistic understanding, guiding manual modifications, and reverse engineering solutions.**

Much recent work has focused on DNN interpretability, and rapid progress has thus far made a thorough systematization of methods difficult. In this survey, we review over 300 works with a focus on inner interpretability tools. We introduce a taxonomy that classifies methods by what part of the network they help to explain (weights, neurons, subnetworks, or latent representations) and whether they are implemented during (intrinsic) or after (post hoc) training. To our knowledge, we are also the first to survey a number of connections between interpretability research and work in adversarial robustness, continual learning, modularity, network compression, and studying the human visual system. We discuss key challenges and argue that the status quo in interpretability research is largely unproductive. Finally, we highlight the importance of future work that emphasizes diagnostics, debugging, adversaries, and benchmarking in order to make interpretability tools more useful to engineers in practical applications.

***Index Terms*—interpretability, explainability, transparency**

I. Introduction

A defining feature of the last decade of deep learning is drastic increases in scale and capabilities [143], [263], with the training compute for machine learning systems growing by ten orders of magnitude from 2010 to 2022 [262]. At the same time, deep neural networks (DNNs) are increasingly being deployed in the real world. If rapid progress continues, broad-domain artificial intelligence could be highly impactful [39], [59], [208], [226], [245], [276].

Given this potential, it is important that practitioners can understand how AI systems make decisions, especially their issues. Models are most typically evaluated by their performance on a test set for a particular task. This raises concerns because a black box performing well on a test set does not imply that the learned solution is adequate. Testing sets typically fail to capture the full deployment distribution, including potential adversarial inputs. They also fail to reveal problems with a model that do not directly relate to test performance (e.g. learning harmful biases). Moreover, even if a user is aware of inadequacies, the black-box nature of a system can make it difficult to fix issues. Thus, a key step to building safe and trustworthy AI systems is to have an expanded toolbox for detecting and addressing problems.

We define an *interpretability method* as any process by which an AI system's computations can be characterized in human-understandable terms. This encompasses a broad set of techniques in the literature on DNNs, but in this paper, we focus specifically on methods for understanding internal structures and representations (i.e. not data, inputs, outputs, or the model as a whole). We call these *inner* interpretability methods. We introduce a taxonomy for these methods, provide an overview of the literature, highlight key connections between interpretability and other topics in deep learning, and conclude with directions for continued work. Our central goals are twofold: (1) to provide a thorough resource for existing inner interpretability work and (2) to propose directions for continued research.

A. The Importance of Interpretable AI

Here, we outline several major motivations.

Open-ended evaluation: Short of actually deploying a system, any method of evaluating its performance can fundamentally only be a proxy for its performance. In particular, test sets can fail to reveal–and often incentivize–undesirable solutions such as dataset bias, socially harmful biases, or developing deceptive solutions. Thus, it is important to have additional ways of rigorously evaluating systems' performance. One of the most important advantages of interpretability techniques lies in their unique ability to, unlike standard evaluation methods, allow humans to more open-endedly study a model and search for flaws.

Showcasing failure: Uncovering why a model fails to produce a correct output can offer insights into what failures look like and how to detect them. This can help researchers avoid issues and help regulators establish appropriate rules for deployed systems.

Fixing bugs: By understanding a problem and/or producing examples that exploit it, a network can be redesigned, probed,

∗ Equal contribution

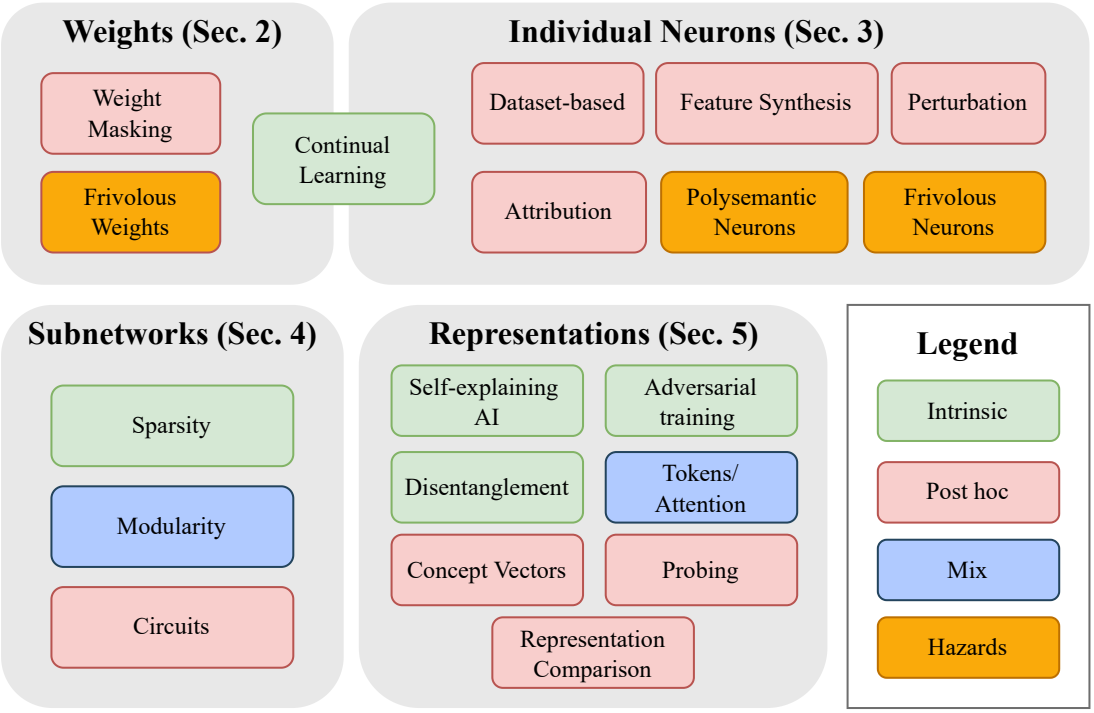

Fig. 1. A taxonomy of inner interpretability methods and hazards associated with them. This mirrors our organization of Sections II-V. We organize methods first by what part of the network's computational graph they help to explain: **weights**, **neurons**, **subnetworks**, or **latent representations**. Second, we organize approaches by whether they are **intrinsic** (implemented during training), **post hoc** (implemented after training), or can rely on a **mix** of intrinsic and post hoc techniques. Finally, we also discuss prominent **hazards** for different approaches.

fine-tuned, and/or adversarially-trained to better align it with the user's goals.

Determining accountability: Properly characterizing behavior is essential for establishing responsibility in the case of misuse or deployment failures.

Improvements in basic understanding: By offering users more basic insights on models, data, and/or algorithms, interpretability techniques could be useful for reducing risks in deployed systems or better forecasting progress in AI. However, improved basic understanding could also be harmful if it causes advancements in risky capabilities to outpace effective oversight. We discuss this in Section VII.

"Microscope" AI: Rigorously understanding how an AI system accomplishes a task may provide additional domain knowledge. This could include insights about solving the task as a whole or the properties of specific examples. This goal has been referred to as "microscope" AI [133], and it could allow for reverse engineering more understandable or verifiable solutions. This may be especially valuable for studying systems with superhuman performance.

B. Scope

Inner Interpretability: Our focus is on *inner* interpretability methods for DNNs. Black-box techniques, adversarial techniques, input attribution methods, neurosymbolic methods, and "good old-fashioned AI" are all valuable but beyond the scope of this survey. This is not to say that they are of less value to building safer AI than the methods we focus on – many of them have major advantages, and a diverse interpretability toolbox is important. However, we focus on inner interpretability methods because (1) there is a great deal of current interest in them and (2) they are well-equipped for certain goals such as guiding manual modifications, reverse engineering solutions, and detecting inner "latent" knowledge which may contribute to deceptive behavior.

Contrasts with past survey works: See also several previous surveys and critiques of interpretability work that overlap with ours [3], [69], [71], [79], [112], [136], [155], [178], [200], [202]–[204], [242], [243], [250], [253], [254], [313]. Unlike this survey, [79], [155], [178], [200], [242] are critique/position papers that do not extensively survey existing work, [3], [71], [112], [203], [204], [243], [253], [254] focus mostly or entirely on approaches out of the scope of this work (e.g. non-DNNs or feature attribution), [69], [136] only survey methods for language models, [313] only focuses on convolutional networks, [250] only surveys single-neuron methods, [203], [253] only focus on post-hoc methods, and [3], [79], [112], [178], [200], [242], [254], [313] are relatively old (before 2020). This survey is also distinct from all of the above in its focus on inner interpretability, implications for diagnostics and debugging, and the intersections between interpretability and a number of other research paradigms.

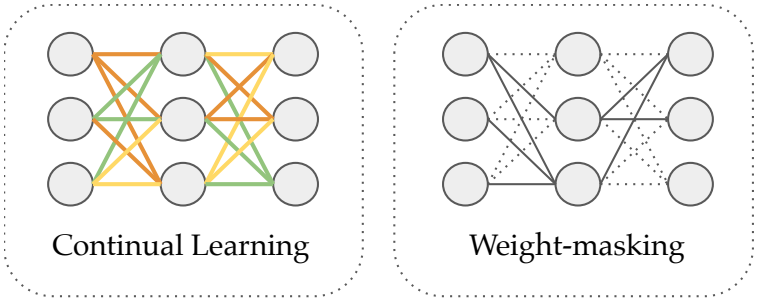

Fig. 2. Inner interpretability methods for weights can focus on (1) **continual learning** techniques that make weights specialize in particular tasks or (2) **weight-masking** techniques which learn a mask over weights as a way of discovering which weights are key for a certain task.

C. Taxonomy

Our taxonomy divides inner interpretability techniques by what part of the DNN's computational graph they explain: **weights**, **neurons**, **subnetworks**, or **latent representations**. We dedicate Sections II-V respectively to each of these approaches. Interpretability techniques can also be divided by whether they are used during or after training. **Intrinsic** interpretability techniques involve training models to be easier to study or come with natural interpretations. **Post hoc** techniques aim to interpret a model after it has been trained. We divide methods by whether they are intrinsic or post hoc at the subsection level.

Fig. 1 depicts our taxonomy and previews the organization of Sections II-V. Note that this taxonomy sometimes divides related methods. For example, continual learning methods for weights (Section II-A) and neurons (Section III-A) are conceptually similar, and methods for interpreting subnetworks (Section IV) frequently involve variations or applications of methods for weights II) or neurons III). We note these connections as we discuss the families of methods below. However, we divide methods first by what part of the network they target because how a technique *operates* on a network typically matters more for goal-oriented engineering than whether it occurs during or after training.

II. Weights

A. Continual Learning (Intrinsic):

One research paradigm in deep learning is to train systems that can learn new tasks without forgetting old ones. This is known as *continual learning* or avoiding *catastrophic forgetting* [72], [266]. Some techniques are based on the principle of having weights specialize for particular types of input data, updating more for some than others [8], [11], [150], [175], [191], [277], [311]. This offers a natural way to characterize weights based on the tasks or classes that they specialize in. Unfortunately, current research on these methods has not been done with an emphasis on improving interpretations of weights or subnetworks. This may be a useful direction for future work. See also methods for continual learning that operate on neurons in Section III-A.

B. Weight-Masking (Post Hoc):

In contrast to intrinsic methods, one can also train weight masks over a network to determine which weights are essential for which tasks [65], [303], [318]. For example, a mask over a classifier's weights can be trained to cover as many as possible while preserving performance on a subset of data. The resulting mask identifies a subset of weights (and a corresponding subnetwork) that can be causally understood as specializing in that subtask. This approach also works for identifying subnetworks that specialize in a task (Section IV).

C. Frivolous Weights (Hazard):

A difficulty in interpreting weights is that many are often unimportant to the network. Past works have shown that networks can often be pruned to contain a very small fraction of their original weights with little to no loss in performance (though sometimes with fine-tuning) [33], [102], [282]. See also frivolous neurons (Section III-G).

III. Individual Neurons

As is common in the literature, we use "neuron" to refer both to units in dense layers and elements of feature maps in convolutional layers.

A. Continual Learning (Intrinsic):

Just as continual learning [72], [266] can be facilitated via specialization among weights (see Section II-A), the same can be done with neurons. Unlike weight-based continual learning methods, which have weights update more for some tasks than others, neuron-based ones typically rely on adding new neurons to the architecture upon encountering a new task [167], [246], [309]. This discourages neurons from learning to simultaneously detect features from multiple unrelated tasks and allows for natural interpretations of neurons in terms of what subtasks they specialize in. As with continual learning methods that operate on weights, current research on these methods has not been done with an emphasis on improving interpretations of neurons or subnetworks. This may be a useful direction for future work. See also Section IV-B, which discusses methods for modularity among neurons.

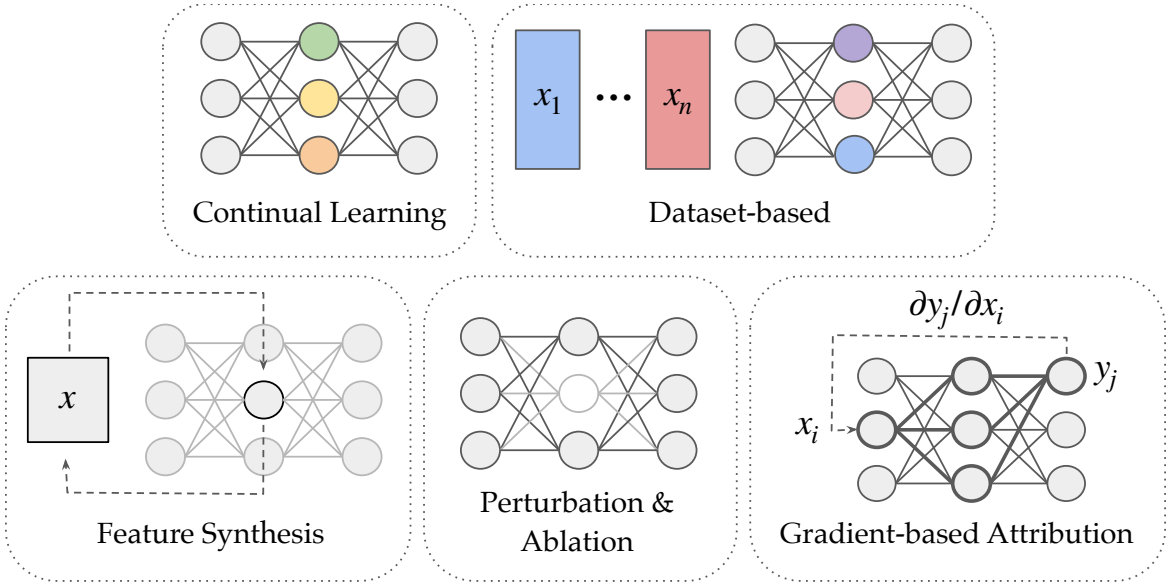

Fig. 3. Inner interpretability methods for individual neurons can focus on (1) **continual learning** techniques that make neurons specialize in particular tasks, (2) **dataset-based** techniques that aim to find which neurons respond to which features, (3) **feature synthesis** to construct inputs that excite individual neurons, (4) **perturbation or ablation** of neurons coupled with analysis of changes to network behavior, and (5) **gradient-based attribution** methods that analyze partial derivatives of outputs w.r.t. neural activations.

B. Dataset-Based (Post Hoc):

A simple way to characterize the role of individual neurons is to use a dataset to analyze which types of inputs they respond to. Perhaps the simplest example of this is searching through a dataset and selecting the inputs that maximally excite a given neuron [322]. A more sophisticated technique known as network "dissection" uses a richly-labeled dataset of semantic concepts to analyze neural responses [25]–[27]. A neuron can then be characterized based on how well its activations align with different types of input. This line of work has been extended to assign descriptions to neurons using compositional logic expressions on a set of labels [207]. This allows the interpretability of neurons to be quantified as the intersection over union for a logical formula on input features and the neuron's activations. This has been further extended to develop natural language explanations by using captioning methods to describe a set of image patches that activate a neuron [125], [218]. These approaches have proven useful for identifying undesirable biases in networks [125], [207]. Dissection has also been used to analyze what types of neurons are exploited by adversarial examples [305], identify failure modes for text-to-image models [57], and probe neural responses in transformers to isolate where specific information is stored [68], [98], [105], [106], [197], [287]. This can then be followed by improving the model by editing a learned fact (such as an undesirable bias) [68], [197]. Unfortunately, all dataset-based methods are limited by the diversity of examples in the dataset used and the quality of labels. See also probing methods in Section V-F.

C. Feature Synthesis (Post Hoc):

This approach is based on synthesizing inputs with the goal of maximally (or minimally) activating a given neuron or combination of neurons. Synthesis methods come with the advantage of not being limited to a particular dataset. Several works have taken this approach, optimizing inputs to excite particular neurons [189], [212], [222]. One can use a distance measure in the optimization objective to synthesize a batch of inputs to be diverse [222]. There has also been work on using generative models instead of directly optimizing over input features [49], [51], [210], [211]. A broader survey of these types of methods is provided by [213]. However [38] finds that natural exemplars which strongly-activate individual neurons can be more useful for helping humans predict neural responses to data than synthesized features.

D. Neural Perturbation and Ablation (Post Hoc):

By analyzing a DNN's behavior under perturbation to a neuron, one can gain insight into the type of information it processes. For example, if a neuron in an image classifier robustly and uniquely detects dogs, one should expect performance on dog classification to worsen when that neuron is ablated (i.e. dropped out). A key benefit of these methods is that they allow for testing counterfactuals, helping establish a causal rather than a correlational relationship between neural activations and the behavior of the network. Works in this area have used neural ablation [127], [324], subspace ablation [206], [235], and non-ablation perturbations [24], [84]. Notably, the net effect of perturbing a neuron can vary by context and which others, if any, are also perturbed. To account for this, one can compute Shapley values for neurons to measure their importance averaged over the ablation of other neurons [111],

[269], and this has been shown to be a practical way of identifying neurons that can be removed or modified to reduce bias or improve robustness [111]. Shapley values, however, are limited in their ability to provide useful causal explanations [159].

E. Gradient-Based Attribution (Post Hoc):

Much work has been done on gradient-based feature attribution to study which features are influential for neural responses or model outputs. There are several surveys and critiques of feature attribution methods in particular [4], [5], [15], [74], [77], [99], [129], [139], [141], [215], [265], [317]. Most of this work has been done to study attributions on *inputs* and is outside the scope of this survey. However, the same type of approach has been applied for attribution with internal neurons. [272] introduce an approach for this using gradients along with runtime tests for sensitivity and invariance to evaluate the quality of interpretations. Building off this, several works have found gradient-based attribution useful in large language models [14], [81], [187], including to guide a search for where certain facts are stored [68]. However, these methods are limited in that explanations are only as valid as the local linear approximation the gradient is based on, and they cannot directly provide causal explanations.

F. Polysemantic Neurons (Hazard):

Polysemantic neurons are activated by multiple unrelated features. They have been discovered via dataset-based methods [100], [125], [207], various forms of visual feature synthesis [114], [212], [220], [289], and feature attribution [85]. How and why they form remains an open question. However, [220] observed a tendency for monosemantic neurons to become polysemantic over the course of training and hypothesized that it is associated with representing information more efficiently. This would suggest that polysemantic neurons might be useful for model performance. However, they also pose a significant challenge for two reasons. First, interpretations of polysemantic neurons are more likely to be incorrect or incomplete. Second, it has been shown that they can be exploited for adversarial attacks [125], [207]. See also Section V-C for a discussion of *entanglement*, which generalizes the notion of polysemanticity to layers.

G. Frivolous Neurons (Hazard):

Frivolous neurons are not important to a network. [48] defines and detects two distinct types: *prunable* neurons, which can be removed from a network by ablation, and *redundant* neurons, which can be removed by refactoring weight matrices. They pose a challenge for interpretability because a frivolous neuron's contribution to the network may be meaningless or difficult to detect with certain methods (e.g., neural perturbation). Network compression may offer a solution. For example, [122], [131], [188], [249], [267] each compress networks by eliminating frivolous neurons. Compression and the interpretability of neurons are linked. After compressing a network, [174] found that the remaining neurons were more interpretable with only marginal change in performance, and [307] used proxies for neuron interpretability to guide neuron-level pruning. Additionally, the motivation for pruning to increase interpretability is closely-related to intrinsically interpretable layer representations. See also Section II-C on frivolous weights and Section V-C on neural disentanglement.

IV. Subnetworks

Note that many of the methods used for analyzing subnetworks supervene on techniques for weights (Section II) or neurons (Section III).

A. Sparsity (Intrinsic):

Sparse weights inside of DNNs allow for much simpler analysis of relationships between neurons. In some cases, sparsification can reduce the number of weights by almost two orders of magnitude while causing little to no tradeoff with performance [102]. Sparsity-aided interpretability has been explored through pruning [29], [101], [205], [302] regularization [241], and sparse attention [196]. In particular, [302] demonstrates how sparsity can be paired with post-hoc techniques for neuron analysis to help a human to edit a model. This has direct implications for safety and debiasing. Pruning portions of the network architecture can also be guided by measures of interpretability [295], [308]. Meanwhile, as an alternative to conventional sparsity, [304] introduce a method to regularize the behavior of a neural network to mimic that of a decision tree.

While sparsity simplifies the analysis of subnetworks, it may not improve the interpretability of *individual neurons*. [101] find no increase in their interpretability through the dissection of pruned networks, and [196] fail to find evidence of improved interpretability of individual neurons with sparse attention.

B. Modularity (Intrinsic):

Modularity is a common principle of engineered systems and allows for a model to be understood by analyzing its parts separately. At a high level, [13] offers a survey of DNN modularization techniques, and [7], [201] study the capabilities and generality of modular networks compared to monolithic ones. The simplest way to design a modular DNN is to use an explicitly modular architecture. This can be considered "hard" modularity. This can be a form of "model-aided deep learning" [264] if domain-specific considerations are used to guide the design. Modular architectures were studied by [290] who analyzed the extent to which neurons in a branched architecture learned to process different features from those in other branches, and [306] who experimented with distinct neural modules that were trained to execute algorithmic subroutines.

Aside from branched architectures, a "softer" form of modularity can be achieved if neurons in different modules are connected to each other but must compete for access to information. This can allow for end-to-end differentiability,

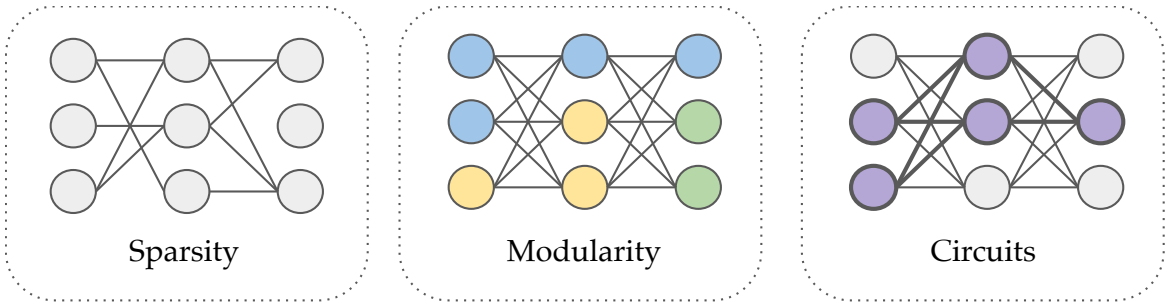

Fig. 4. Inner interpretability methods for subnetworks can focus on (1) simplifying the computational subgraph via **sparsity**, (2) either intrinsic methods to enforce **modularity** among neurons or post hoc methods to group them into modules, or (3) analysis of neural **circuits** which can be understood as performing a specific task.

yet sparse information flow between modules. Methods for soft modularity have been studied via initialization [96], regularization [96], a controller [140], [151], or sparse attention [16], [85], [115], [260]. Notably [260] used attention to both induce specialization among neurons and reduce catastrophic forgetting. See also methods for avoiding catastrophic forgetting by having subsets of neurons specialize in a given task in Section III-A.

C. Modular Partitionings (Post Hoc):

A post hoc way of understanding a DNN in terms of modules is to partition the neurons into a set of subnetworks, each composed of related neurons. Toward this goal, [96], [297], [298] divide neurons into modules based on graphical analysis of the network's weights and analyze how distinct the neurons in each module are. These methods involve no data or runtime analysis. In contrast, [18], [50], [127], [163], [296] each perform partitioning and cluster analysis based on how neurons associate with inputs and/or outputs. In particular, [127] present a statistical pipeline for estimating the interpretability of neuron clusters without a human in the loop. Overall, however, these methods have had very limited success in finding highly-composite partitionings in models. A useful direction for future work may be to combine this approach with intrinsic modularity methods.

D. Circuits Analysis (Post Hoc):

Instead of analyzing an entire partition of a network, a much simpler approach is to study individual subnetworks inside of it. These have often been referred to as neural "circuits" which can be as small as just a few neurons and weights. This has been done with weight masking [65], [294], data-based methods [94], [256], [279], [314], [315], feature synthesis [45], [219]–[223], [232], [259], [289], and neural ablation [118], [198]. However, many of the successes of circuits analysis to date have focused on toy models and involved intensive effort from human experts. To be useful for improving models in practical applications, future methods will likely need to leverage automation. [238] make progress on this by distilling a DNN into a sparse symbolic causal graph on a set of concepts and providing theoretical guarantees for the faithfulness of the graph. See also Section V-D for a discussion of circuits in transformers.

V. Internal Representations

A. Self-Explaining Models (Intrinsic):

Most methods in the literature used for understanding DNNs aim to help a human "open up" the network and study parts of it. If one wants to understand another human's reasoning, the analogous techniques would involve studying their brain directly. These are sometimes useful, but in most cases, simply asking another human for an explanation of what they are thinking is much more effective. Self-explaining AI systems are meant to provide such explanations of internal reasoning in an analogous way to how humans can provide them. Competing definitions are offered in the literature, but we will use one based on [87], which simply requires that a model produces an explanation for its reasoning that can be easily be understood by a human, ideally paired with a confidence estimate.

In computer vision, one approach has been to classify images based on their similarity to a set of learned "prototypes" [12], [53], [145], [171], [247], [312]. Prototype-based classification has also been studied in language models [92]. These methods allow the model to attribute its outputs to a set of exemplary datapoints, allowing its decision to be explained as "*this* input resembles *these* other examples."

Another self-explaining AI strategy has been to supervise human-understandable explanations for model outputs that are computed from the same inner representations. In computer vision, this has been done for classification and question answering [9], [123], [124], [149], [229]. In natural language processing, this has been done for question answering and natural language inference with explanations [43], [160], [162], [319]. For large language models that have sufficiently general language capabilities, explanations can also simply be elicited via prompts (e.g., [40], [58]). However, the extent to which these explanations accurately explain the model's decision making is very unclear [142].

[12] argues that explanations should meet three criteria: (1) Explicitness: are explanations direct and understandable? (2) Faithfulness: do they correctly characterize the decision? And

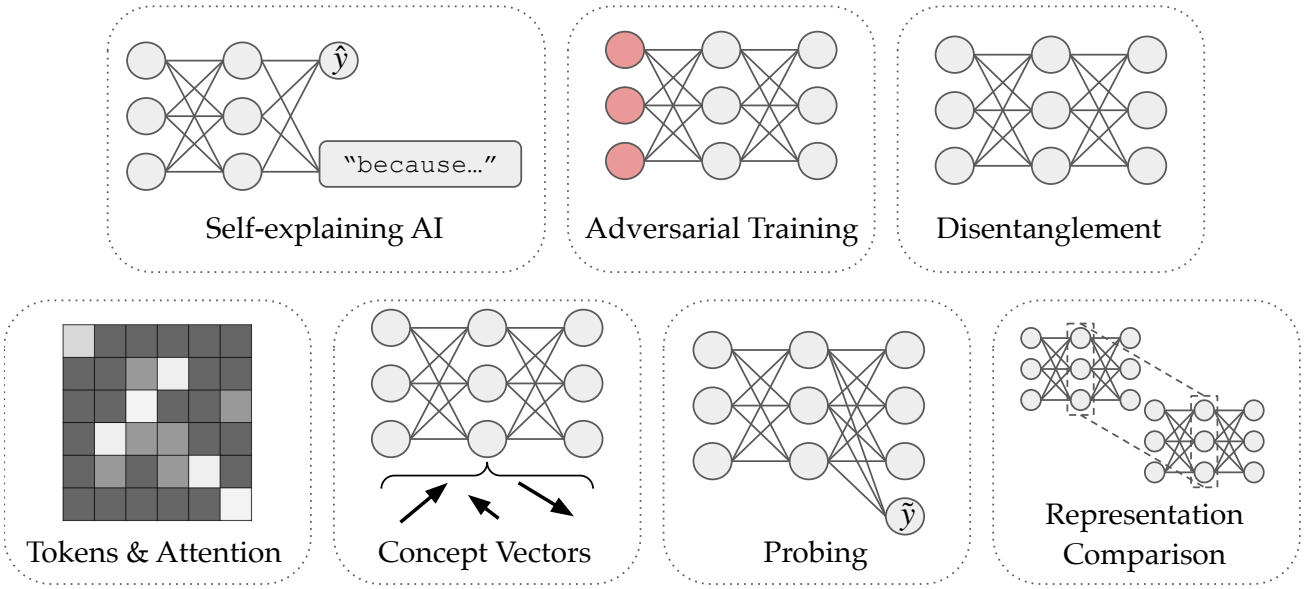

Fig. 5. Inner interpretability methods for neural representations can focus on (1) training **self-explaining AI** systems that explain their decisions, (2) **adversarial training**, (3) the **disentanglement** of latent representations such that each neuron tends to uniquely respond to a single concept in data, (4) analysis of **token** evolution or **attention** maps in transformers, (5) analysis of **concept vectors** in latent space, (6) **probing** neural representations to evaluate their transferability to a target task, and (7) **representation comparison** between different layers in two networks.

(3) Stability: how consistent are they for similar examples? It has been shown that explanations from such models can be unfaithful [12], [283] or vulnerable to adversarial examples [44], [128], [320], so producing self-explaining models that meet these remains an open challenge. Toward fixing these issues, [75] introduces an NLP benchmark, and [36] provides an interactive debugging method for prototype networks.

B. Adversarial Training (Intrinsic)

[88] found that adversarially trained classifiers exhibited improvements in a number of interpretability-related properties, including feature synthesis for neurons (see Section III-C). It has also been found that these adversarially trained networks produce better representations for transfer learning [252], image synthesis [49], [51], [255], and for modeling the human visual system [89]. Unfortunately, robustness may be at odds with accuracy [281], potentially due to predictive but "nonrobust" features in a dataset [135]. This had led to an understanding that adversarial examples can be used to help to understand what types of useful or exploitable features a network detects and represents [49], [51].

C. Disentanglement (Intrinsic):

During a pass through a network, each layer's activations can be represented as a vector in latent space. The goal of *disentanglement* [30] is ensuring that features can be more easily identified from studying a latent vector by encouraging a more bijective relationship between neurons and a set of interpretable concepts. See also Section III-F for a discussion of polysemantic neurons. Disentanglement can be done in a supervised manner by encouraging neurons to align to a set of predetermined concepts. [56] did this by applying a whitening operation to decorrelate features followed by a learned orthogonal transformation to produce latent activations that could be supervised. Similarly, inner supervision was used by [152], [183], [184] to train a 'bottleneck' layer to separate features, and by [271] to learn sparse, interpretable embeddings. However, [190] discusses challenges with these models, particularly the problem of "leakage" in which undesired information nonetheless makes it through the bottleneck.

Disentanglement can also be done in an unsupervised manner. A partial example of this is dropout [268] which prevents co-adaptation among neurons, though at the cost of increasing redundancy. Other works have explored using lateral inhibition between neurons in a layer to make them compete for activation [46], [85], [157], [271], designing a "capsule" based architecture in which a group of neurons have activations that each represent a specific feature [73], [248], aligning activations to components of variation in data [161], using a mutual information loss [55], using an inter-class activation entropy-based loss [316], regularizing the Hessian of the network's outputs w.r.t a layer [230], training a classifier and autoencoder from the same latents [258], or learning a mask over features [121]. Other works have focused specifically on autoencoders, training them to have more independently-activated neurons [41], [54], [126], [148], [158]. However, in a survey of these methods [181], [182] prove an impossibility result for unsupervised disentanglement without inductive biases on both the models and data.

D. Tokens and Attention (Intrinsic and Post Hoc):

Transformer architectures process data by passing token representations through attention and feed forward layers in alternating fashion. These architectural building blocks pose

unique opportunities for studying the network's internal representations.

First, the tokens can be studied. This can be done by interpreting token representations in transformers directly [84], [108], [109], [169], [209], [224] or analyzing how fully-connected layers process them [110], [209].

Second, key-query products are computed inside of an attention layer and represent how much each inner token is attending to others. This notion of studying relations between token representations has similarities to circuits analysis covered in Section IV-D. In their seminal work, [20] showed that an attentional alignment appeared to show the expected attention patterns for translation. Other recent works have used this approach more systematically [2], [62], [119], [284] including for the study of harmful biases [72]. And [239] introduced a "ConceptTransformer" whose outputs can be explained as an attention map over user-defined concepts much like a concept bottleneck network [152]. Interactive tools for visual analysis of attentional attribution are provided by [166], [179], [270], [286]. And [52], [84], [224] expanded on this approach toward the goal of multi-step attribution across multiple layers. Importantly, the analysis of attention may not always suggest faithful explanations, and an over-reliance on them for interpretation can be hazardous [138], [261], [300]. Finally, transformers may have many frivolous, prunable attention heads [288], suggesting a further need for caution because not all heads may be worth interpreting.

E. Concept Vectors (Post Hoc):

While disentanglement aims to align concepts with individual neurons, methods for analyzing concept vectors are post hoc solutions to the same problem. Here, the goal is to associate directions in latent space with meaningful concepts. Several works have done this by analysis of activations induced by images from a dataset of concepts [100], [146], [185], [186], [237], [323] including [1], [310] who used it explicitly for debugging. A contrasting approach was used by [258]. Rather than beginning with concepts and then identifying directions for them, they first identified directions using a generator and a "layer selectivity" heuristic, and then sought to find post hoc explanations of what they encoded. A debugging-oriented approach was taken by [137], [301] who classified and clustered embeddings of data examples that were incorrectly labeled by a classifier, including cases due to demographic biases. This allowed for detection, interpretation, and intervention for potentially difficult inputs for the model as well as a way to identify underrepresented subcategories of data. Unfortunately for these approaches, there is evidence that networks learn to represent many more useful concepts than can linearly independently be represented by their internal layers [86]. And single directions in activation space can correspond to unrelated concepts depending on the activation vector's magnitude [32].

F. Probing (Post Hoc):

Given some way of embedding data, the goal of probing is to understand whether or not that embedding captures a certain type of information. Probing leverages transfer learning as a test for whether embeddings carry information about a target task. The three steps to probing are to (1) obtain a dataset that contains examples capturing variation in some quality of interest, (2) embed the examples, and (3) train a model on those embeddings to see if it can learn the quality of interest. Any inner representation from any model can be used, making this a versatile technique. A survey of probing works is provided by [28]. The simplest example of probing is to use an unsupervised learning algorithm as the probe [130]. Additional work has been done with linear probes for image classifiers [10]. However, probing has most commonly been done in language models [6], [63], [91], [117], [153], [168], [170], [177], [199], [216], [231], [251], [273]. Notably, a form of contrastive probing was used by [42] for detecting deception in language models. While versatile, probing is imperfect [17]. One issue is that a probe's failure to learn to represent the desired quality in data is not necessarily an indicator that it is not represented. For example, this may be the case with an underparameterized probe. On the other hand, a successful probe does not necessarily imply that the model being probed actually uses that information about the data. This was demonstrated by [236] who argued for the use of rigorous controls when probing. In a subsequent paper, [83] aimed to address this problem by pairing probing with experiments that manipulated the data in order to analyze the causal influence of perturbations on performance. See also dataset-based methods for characterizing neurons in Section III-B.

G. Representation Comparison (Post Hoc):

A somewhat indirect way of characterizing the representations learned by a DNN is to estimate the similarity between its inner representations and those of another DNN. This is challenging to quantify because networks are highly nonlinear and represent concepts in complex ways that may not reliably align with neurons or activation vectors. Nonetheless, a set of works have emerged to address this problem with a variety of both linear and nonlinear methods. These include single-neuron alignment [113], [172], [173], [275], vector-space alignment [293], canonical correlation analysis [206], singular vector canonical correlation analysis [233], centered kernel alignment [154], [214], [234], [274], deconfounded representation similarity [66], layer reconstruction [176], model stitching [21], [64], [194], representational similarity analysis [195], representation topology divergence [22], and probing [93]. Methods like these may aid in a better basic understanding of what features networks learn and how. However, different methods often disagree about the extent to which layers are similar. [76] argue that these methods should be sensitive to changes that affect functional behavior and invariant to ones that do not. They introduce a benchmark for evaluating similarity measures and show that two of the most common

methods, canonical correlation analysis and centered kernel alignment, each fail in one of these respects.

VI. Discussion

Interpretability is closely linked with adversarial robustness research. There are several connections between the two areas, including some results from non-inner interpretability research. (1) More interpretable DNNs are more robust to adversaries [141]. A number of works have studied this connection by regularizing the input gradients of networks to improve robustness [37], [80], [90], [97], [120], [144], [147], [192], [217], [240], [257]. Aside from this, [82] use lateral inhibition and [280] use a second-order optimization technique, each to improve *both* interpretability and robustness. Furthermore, emulating properties of the human visual system in a convolutional neural network improves robustness [70]. (2) More robust networks are more interpretable [19], [88], [227]. Adversarially trained networks also produce better representations for transfer learning [7], [252], image generation [49], [51], [255], and modeling the human visual system [89]. (3) Interpretability tools can be used to design adversaries. Doing so is a way to rigorously demonstrate the usefulness of the interpretability tool. This has been done by [47], [49], [51], [125], [207] and has been used to more effectively generate adversarial training data [326]. As a means of debugging models, [132] argues for using "relaxed" adversarial training, which can rely on interpretability techniques to discover distributions of inputs or latents which may cause a model to fail. (4) Adversarial examples can be interpretability tools [49], [51], [78], [135], [278] including adversarial trojan detection methods [104], [116], [180], [291], [292], [321].

Interpretability is also closely linked with continual learning, modularity, network compression, and semblance to the human visual system. Continual learning methods involving parameter isolation, and/or regularization make neurons and weights more intrinsically interpretable. Sections II-A and III-A discussed how these methods suggest intrinsic interpretations for weights and/or neurons. Thus, they allow for each weight or neuron to be understood as having partial memberships in a set of task-defined *modules*. Aside from this, a number of other intrinsic modularity techniques were the focus of Section IV-B. And as discussed in Section IV-C, networks can also be interpreted by partitioning them into modules and studying each separately. Moreover, "frivolous" neurons, as discussed in Section III-G, can include sets of redundant neurons which can be interpreted as modules. Networks with frivolous neurons are compressible, and compression can guide interpretations, and interpretations can guide compression, as discussed in Section III-G.

Interpretability techniques should scale to large models. Small networks and simple tasks such as MNIST classification [165] are often used for testing methods. However, simple networks performing simple tasks can only be deployed in a limited number of real world settings, and they are sometimes easy to replace with other intrinsically interpretable, non-network models. As a result, the scalability of a technique is strongly related to its usefulness. For example, capsule networks [248] achieve impressive performance on MNIST classification and have intrinsic interpretability properties that convolutional networks lack. However, they are much less parameter efficient and have thus far not achieved competitive performance beyond the CIFAR-10 [156] level, let alone the ImageNet [244] level [228]. Methods like these may offer excellent inspiration for future work, but if they fail to be tractable for large models, they will be of limited direct value for practical interpretability. We urge researchers to detail computational requirements and test the scalability of their methods.

Interpretability techniques generate hypotheses – not conclusions. Producing merely-plausible explanations is insufficient. Evaluating validity and uncertainty are key. Mistaking hypotheses for conclusions is a pervasive problem in the interpretability literature [178], [200], [242]. Consider the goal of explaining a particular neuron. There exist several methods to do so (Section III). However, if such an approach suggests that the neuron has a particular role, this does not offer any guarantee that this explanation is complete and faithful to its true function. Often, very plausible-seeming explanations do not pass simple sanity checks [4] or are very easy to find counterexamples for [35], [128], [220]. A great number of works in interpretability have failed to go beyond simply inspecting the results of a method. More care is needed. Interpretability techniques can only be evaluated to the extent that they help users make testable predictions. They can only genuinely be useful inasmuch as those predictions validate. And the validity of an interpretation is only granted on the distribution of data for which validating tests were conducted – extrapolating interpretations is risky (e.g., [35]). Developing specific methods for evaluating interpretability techniques is discussed later in Section VII.

In addition to validity, it is important to quantify uncertainty. Ideally, interpretations should be paired with confidence estimates. How to measure certainty depends on the method at hand, but some approaches have been used such as supervising explanations (e.g., [123]), conducting multiple trials (e.g., [220]), comparisons to random baselines (e.g., [127], [236]), comparisons to other simple methods [4]. or searching for cases in which an interpretation fails (e.g., [25], [35], [128]).

Cherry-picking is harmful and pervasive. Evaluation of methods should not fixate on best-case performance. Due to the inherent difficulty of interpreting DNNs, many works in the literature showcase individual, highly successful applications of their method, often in toy models. This can be useful for providing illustrative examples or specific insights. But the evaluation of interpretability techniques should not be biased toward their best case performance. One hazard of doing this could be from overestimating the value of techniques. And in fact, some works have found that certain methods only tend to

perform well on a fraction of examples (e.g., [25], [35], [45], [48], [85], [127], [181], [182], [196], [197], [225], [288]).

Another harm of cherrypicking might come from biasing progress in interpretability toward methods that fail to explain *complex* subprocesses. Some methods are better equipped for this than others. For example, attributing a feature's representation to a linear combination of neurons is strictly more general than attributing it to a single neuron. It is likely that some kinds of features or computations in DNNs are more naturally human-understandable than others, so methods that are only useful for explaining simple subprocesses may be poorly-equipped for studying networks in general.

Works should evaluate how their techniques perform on randomly or adversarially sampled tasks. For example, a work on characterizing neural circuits should not focus only on presenting results from circuits that were particularly amenable to interpretation. It should also aim to explain the role of randomly or adversarially sampled neurons inside of circuits or find circuits that can explain how the network computes randomly or adversarially selected subtasks. If a method like this only succeeds in limited cases, this should be explicitly stated.

Ideally, progress in interpretability should neither decrease performance in general nor increase certain risky capabilities. On one hand, interpretable AI techniques should maintain competitiveness. It is key to avoid costs such as degraded task performance, increases in bias, higher compute demands, or difficulty to use in modern deep learning frameworks. Competitive shortcomings like these could lead to "value erosion" [67] in which safer, more interpretable AI practices are not adopted in favor of more competitive approaches.

On the other hand, certain types of performance improvements from interpretability research may also be undesirable. Interpretability work should also not lead to increased capabilities if they make safety-related oversight more difficult. For example, advances toward general intelligence may lead to serious harms if not managed appropriately [39], [59], [208], [226], [245], [276]. One risky possibility is if interpretability is a byproduct of increased general capabilities. For example, large language models can often be prompted to "explain" their reasoning, but only as a result of having advanced, broad-domain abilities. Another way for this to occur is if interpretability leads to advancements in capabilities via basic model insights. From the perspective of avoiding risks from advanced AI systems, neither of these is ideal. A focus on improving interpretability techniques *without* commensurate increases in capabilities offers the best chance of preventing advancements in AI from outpacing our ability for effective oversight. From this perspective, we argue that improvements in safety rather than capabilities should be the principal goal for future work in interpretability.

VII. Future Work

The connections between interpretability, modularity, adversarial robustness, continual learning, network compression, and semblance to the human visual system should be better understood. One of the most striking findings of modern interpretability work is its connections with other paradigms in deep learning. One of the central goals of this survey has been to highlight these connections (see Section VI). Currently, the intersections in the literature between interpretability and these other areas are relatively sparse. Moving forward, an interdisciplinary understanding of interpretability may lead to insights and advancements spanning multiple domains.

Scaling requires efficient human oversight. Many explanations obtained by state of the art interpretability techniques have involved a degree of human experimentation and creativity in the loop. In some cases, many hours of meticulous effort from experts have been required to explain models or subnetworks performing very simple tasks (e.g., [45], [209]). But if the goal is to obtain a thorough understanding of large systems, human involvement must be efficient. Ideally, humans should be used for screening interpretations instead of generating them. Solutions can include using active learning (e.g., [103]), weak supervision (e.g., [34]), implicit supervision using proxy models trained on human-labeled data (e.g., [49], [51]), and/or rigorous statistical analysis of proxies (e.g., [127], [325]) to reduce the need for human involvement. Toward this end, obtaining additional high-quality datasets (e.g., [25]) with richly-labeled samples may be valuable.

Focus on discovering novel behaviors – not just analyzing them. Many existing methods are only well-equipped to study how models behave in limited settings. For example, any interpretability method that relies on a dataset is limited to characterizing the model's behavior on that data distribution. But ideally, methods should not be limited to a given dataset or to studying potential failures when the failure modes are already known. For instance, an important practical problem is the detection of offensive or toxic speech, but no dataset contains examples of all types of offensive sentences, and having a human hand-specify a function to perfectly identify offensive from inoffensive speech is intractable. Humans can, however, usually identify offensiveness when they see it with ease.

This highlights a need for techniques that allow a user to discover failures that may not be in a typical dataset or easy to think of in advance. This represents one of the *unique* potential benefits of interpretability methods compared to other ways of evaluating models such as test performance. Toward this end, some inner interpretability methods that generate abstract understandings of subnetworks have proven to be useful (e.g., [68], [125], [197], [207], [256]. However, methods based on *synthesizing* adversarial examples may offer a particularly general approach for discovering novel failure modes (e.g., [49], [51], [116], [291]).

Interpretability work may help better understand convergent learning of representations. Some works have hypothesized that similar features or concepts tend to occur across different model instances or architectures [220], [299].

Better understanding the extent to which systems learn similar concepts would lead to a more basic understanding of their representations and how interpretable we should expect them to be. If these hypotheses are true, interpreting one model in depth may be much more likely to lead to generalizable insights. Continued work on measuring representational similarity between neural networks (see Section V-G) may be well-suited for making progress toward this goal.

"Mechanistic interpretability" and "microscope AI" are ambitious but potentially very valuable goals. One direction for interpretability research is *mechanistic interpretability*, which aims to gain an algorithmic-level understanding of a DNN's computations. This can be operationalized as converting the DNN into some form of human-understandable pseudocode [95]. This is related to the goal of *microscope* AI, which refers to gaining domain insights by thoroughly interpreting high-performing AI systems [133]. These capabilities would have advantages, including predicting counterfactual behaviours and reverse engineering models. Thus far, there have been a limited number of attempts towards this goal that have had some success by using small models, simple tasks, and meticulous effort from human experts [45], [85], [209], [285]. Future work in this direction may benefit by using techniques from program synthesis and analysis to automate the generation and validation of hypotheses.

Detecting deception and eliciting latent knowledge may be valuable for advanced systems. A system is *deceptive* if it passes false or incomplete information along some communication channel (e.g., to a human), despite having the capability to pass true and complete information. Relatedly, *latent knowledge* [60] is something that a system "knows" but shows no signs of knowing. For example, a language model might babble a common misconception like "bats are blind" in some contexts despite having the "knowledge" that this is false. Hidden knowledge like this may lead to deceptive behavior. As an example, [60] discusses a system that intentionally and deceptively manipulates the observations that a human sees for monitoring it. In this case, knowledge about the true nature of the observations is latent.

Being able to characterize deceptive behavior and latent knowledge has clear implications for safer highly intelligent AI by allowing humans to know when a model may be untrustworthy. But this may be difficult for several reasons including that (1) by definition, deceptive behavior and latent knowledge cannot be determined by observing the model's deployment behavior alone, (2) any mismatches between the features/concepts used by humans and the model require a method for ontology translation, and (3) it is unclear the extent to which a human can interpret an AI system that is superhuman on some task. However, inner interpretability methods offer a unique approach to these challenges via scrutinizing parts of the model's computational graph that may process latent knowledge. Probing has shown potential for this [42].

Rigorous benchmarks are needed. Ideally, they should measure how helpful methods are for producing *useful* insights that have relevance to engineers. These could involve rediscovering known flaws in networks. Interpretability work on DNNs is done with numerous techniques, not all of which have the same end goal. For example, some methods aim to explain how a DNN handles a single input while others are aimed at a more generalizable understanding of it. For these reasons, plus the rapid development of techniques, widely-accepted benchmarks for interpretability do not yet exist. This may be a limitation for further progress. A well-known example of a benchmark's success at driving immense progress is how ImageNet [244] invigorated work in supervised image classification.

The weakest form of evaluation for an interpretability method is by its ability to merely suggest a particular characterization. For example, if feature synthesis is used to visualize a neuron, having a human look at the visualization and say "this looks like X" is a fraught basis for concluding that the neuron is an X-detector. This would be to conflate a hypothesis with a conclusion. A somewhat more rigorous approach to evaluation is to make a simple testable prediction and validate it. For example, suppose the hypothesized X-detector activates more reliably for inputs that contain X than ones that do not. Another example is if the use of some method improves some quantitative proxy for interpretability (e.g., [127]). This type of approach is valuable but still not ideal.

The end goal for interpretability tools should be to provide valid and useful insights, so methods for evaluating them ought to measure their ability to guide humans in doing *useful* things with models. In other words, interpretability tools should be useful for engineers, particularly ones who want to diagnose and debug models. Some works have made progress toward this. Examples include designing novel adversaries (e.g., [47], [49], [51], [107], [125], [135], [137], [164], [207], [301], [326]), manually editing a network to repurpose it or induce a predictable change in behavior (e.g., [27], [42], [68], [111], [197], [302]), or reverse-engineering a system using interpretability techniques (e.g., [45], [84], [209]).

One tractable approach for benchmarking suggested by [134] would be to evaluate interpretability techniques by their ability to help a human find flaws that an adversary has implanted in a model. Judging techniques by how well they help humans rediscover these flaws would offer a much more direct measure of their practical usefulness than ad hoc approaches to evaluation. Related techniques for feature attribution methods have been argued for [129] and used [5], [23], [74] but have not yet popularized. Competitions for implanting and rediscovering flaws (e.g., [193]) hosted at well-known venues or platforms may be a useful way to drive progress in both techniques and benchmarking.

Combining techniques may lead to better results. Interpretability techniques can often be combined. For example, almost any intrinsic method could be used with almost any post hoc one. However, the large majority of work in inter-

pretability focuses on studying them individually. Studying the interplay between methods is relatively unexplored. Some works have identified useful synergies (e.g., [88], [302]) But to the best of our knowledge, there are no works dedicated to thoroughly studying interactions between different methods. We hope that new baselines and increased demand for rigorously interpretable systems will further incentivize results-oriented interpretability work.

Consider an example. The ImageNet benchmark was very effective at advancing the state of the art on image classification performance in the 2010s. Over this time period, improvements in classification performance were not due to single techniques, but a combination of breakthroughs – batch normalization, residual connections, inception modules, deeper architectures, etc. Similarly, we should not expect to best advance capabilities related to interpretability without a combination of methods.

Applying interpretability techniques for debugging and debiasing in the wild. Working to apply interpretability tools to find issues with real-world models (e.g. [61]) both helps for discovering issues in consequential applications and to test methods to see which ones may be the most practically useful. In doing so, researchers should be critical of the ethical frameworks used in machine learning and particularly how they may diverge from the interests of people–particularly from disadvantaged groups–who may be the most adversely affected by these technologies [31].

Growing the field of interpretability. Many ethical or safety concerns with AI systems can be reduced via tools to better understand how models make decisions and how they may fail. As a result, we argue that instead of being a separate interest, interpretability should be seen as a *requirement* for systems that are deployed in important settings. As discussed above, a compelling path forward is via benchmarking and competitions. There are some reasons for optimism. The field is maturing, and a number of techniques have now proven their worth for practical insights and debugging. And although they are our focus here, we emphasize that that *inner* interpretability methods will not be the only valuable ones for improving AI safety.

It is time for a paradigm shift toward engineering. Currently, the field of interpretability in DNNs is largely unproductive. Being able to rigorously study the solutions learned by DNNs seems to have important potential for making DNNs safer, but they are currently rarely used for evaluation or engineering practical applications. As discussed above, works in the literature too often treat hypotheses as conclusions [178], [200], [242] and fail to connect a method to useful applications. Some amount of exploratory work is clearly valuable for generating insights, and it should continue. But the field has yet to produce many methods that are competitive in real applications.

In large part, the lack of productivity in interpretability seems due to a lack of direction. Motivations for interpretability work are "diverse and discordant" [178] and the term itself, as used in the literature, "lacks precise meanings when applied to algorithms" [155]. We join with [79], [242], [200], and [155] in calling for grounding interpretability in meaningful applications. If interpretability tools are ultimately meant to help engineers diagnose and debug DNNs, the field should design and evaluate methods based on this.

We argue that moving forward, the most pressing change that is needed in the field is a focus on producing tools that are useful to engineers. To better realize the potential of interpretability work for human-aligned AI, a more deliberate, interdisciplinary, and application-focused field will be important. It will be valuable to have more research that emphasizes diagnostics, debugging, adversaries, benchmarking, and leveraging useful combinations of different interpretability tools.

Acknowledgements

We thank Davis Brown, Erik Jenner, Jan Kirchner, Max Lamparth, Ole Jorgensen, Florian Dorner, and Peter Hase for their feedback. Tilman Räuker is supported by the Long-Term Future Fund.

References

[1] Abubakar Abid, Mert Yuksekgonul, and James Zou. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *International Conference on Machine Learning*, pages 66–88. PMLR, 2022.

[2] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[3] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[4] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

[5] Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. *arXiv preprint arXiv:2011.05429*, 2020.

[6] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*, 2016.

[7] Atish Agarwala, Abhimanyu Das, Brendan Juba, Rina Panigrahy, Vatsal Sharan, Xin Wang, and Qiuyi Zhang. One network fits all? modular versus monolithic task formulations in neural networks. *arXiv preprint arXiv:2103.15261*, 2021.

[8] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. *Advances in Neural Information Processing Systems*, 32, 2019.

[9] Zeynep Akata, Lisa Anne Hendricks, Stephan Alaniz, and Trevor Darrell. Generating post-hoc rationales of deep visual classification decisions. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 135–154. Springer, 2018.

[10] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

[11] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019.

[12] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

[13] Mohammed Amer and Tomás Maul. A review of modularization techniques in artificial neural networks. *Artificial Intelligence Review*, 52(1):527–561, 2019.

[14] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

[15] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Gradient-based attribution methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 169–191. Springer, 2019.

[16] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48, 2016.

[17] Omer Antverg and Yonatan Belinkov. On the pitfalls of analyzing individual neurons in language models, 2021.

[18] Sercan Arik and Yu-Han Liu. Explaining deep neural networks using unsupervised clustering. 2020.

[19] Maximilian Augustin, Alexander Meinke, and Matthias Hein. Adversarial robustness on in-and out-distribution improves explainability. In *European Conference on Computer Vision*, pages 228–245. Springer, 2020.

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

[21] Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *Advances in Neural Information Processing Systems*, 34:225–236, 2021.

[22] Serguei Barannikov, Ilya Trofimov, Nikita Balabin, and Evgeny Burnaev. Representation topology divergence: A method for comparing neural network representations. *arXiv preprint arXiv:2201.00058*, 2021.

[23] Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. " will you find these shortcuts?" a protocol for evaluating the faithfulness of input salience methods for text classification. *arXiv preprint arXiv:2111.07367*, 2021.

[24] Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Identifying and controlling important neurons in neural machine translation. *arXiv preprint arXiv:1811.01157*, 2018.

[25] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations, 2017.

[26] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, sep 2020.

[27] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks, 2018.

[28] Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

[29] Gabriel Béna and Dan FM Goodman. Extreme sparsity gives rise to functional specialization. *arXiv preprint arXiv:2106.02626*, 2021.

[30] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[31] Abeba Birhane, Pratyusha Kalluri, Dallas Card, William Agnew, Ravit Dotan, and Michelle Bao. The values encoded in machine learning research. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 173–184, 2022.

[32] Sid Black, Lee Sharkey, Leo Grinsztajn, Eric Winsor, Dan Braun, Jacob Merizian, Kip Parker, Carlos Ramón Guevara, Beren Millidge, Gabriel Alfour, et al. Interpreting neural networks through the polytope lens. *arXiv preprint arXiv:2211.12312*, 2022.

[33] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

[34] Benedikt Boecking, Willie Neiswanger, Eric Xing, and Artur Dubrawski. Interactive weak supervision: Learning useful heuristics for data labeling. *arXiv preprint arXiv:2012.06046*, 2020.

[35] Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. An interpretability illusion for bert, 2021.

[36] Andrea Bontempelli, Fausto Giunchiglia, Andrea Passerini, and Stefano Teso. Toward a unified framework for debugging gray-box models. *arXiv preprint arXiv:2109.11160*, 2021.

[37] Akhilan Boopathy, Sijia Liu, Gaoyuan Zhang, Cynthia Liu, Pin-Yu Chen, Shiyu Chang, and Luca Daniel. Proper network interpretability helps adversarial robustness in classification. In *International Conference on Machine Learning*, pages 1014–1023. PMLR, 2020.

[38] Judy Borowski, Roland S Zimmermann, Judith Schepers, Robert Geirhos, Thomas SA Wallis, Matthias Bethge, and Wieland Brendel. Exemplary natural images explain cnn activations better than state-of-the-art feature visualization. *arXiv preprint arXiv:2010.12606*, 2020.

[39] N. Bostrom. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press, 2014.

[40] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[41] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[42] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

[43] Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31, 2018.

[44] Oana-Maria Camburu, Brendan Shillingford, Pasquale Minervini, Thomas Lukasiewicz, and Phil Blunsom. Make up your mind! adversarial generation of inconsistent natural language explanations. *arXiv preprint arXiv:1910.03065*, 2019.

[45] Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah. Curve detectors. *Distill*, 5(6):e00024–003, 2020.

[46] Chunshui Cao, Yongzhen Huang, Zilei Wang, Liang Wang, Ninglong Xu, and Tieniu Tan. Lateral inhibition-inspired convolutional neural network for visual attention and saliency detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[47] Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Exploring neural networks with activation atlases. *Distill.*, 2019.

[48] Stephen Casper, Xavier Boix, Vanessa D'Amario, Ling Guo, Martin Schrimpf, Kasper Vinken, and Gabriel Kreiman. Frivolous units: Wider networks are not really that wide. *arXiv preprint arXiv:1912.04783*, 2019.

[49] Stephen Casper, Kaivalya Hariharan, and Dylan Hadfield-Menell. Diagnostics for deep neural networks with automated copy/paste attacks. *arXiv preprint arXiv:2211.10024*, 2022.

[50] Stephen Casper, Shlomi Hod, Daniel Filan, Cody Wild, Andrew Critch, and Stuart Russell. Graphical clusterability and local specialization in deep neural networks. In *ICLR 2022 Workshop on PAIR^2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022.

[51] Stephen Casper, Max Nadeau, Dylan Hadfield-Menell, and Gabriel Kreiman. Robust feature level adversaries are interpretability tools. *arXiv preprint arXiv:2110.03605*, 2021.

[52] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021.

[53] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

[54] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

[55] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[56] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

[57] Jaemin Cho, Abhay Zala, and Mohit Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generative transformers. *arXiv preprint arXiv:2202.04053*, 2022.

[58] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[59] Brian Christian. *The alignment problem: Machine learning and human values*. WW Norton & Company, 2020.

[60] Paul Christiano, Ajeya Cotra, and Mark Xu. Eliciting latent knowledge: How to tell if your eyes deceive you. 2022.

[61] Jack Clark, Rob Reich, Marietje Schaake, Rumman Chowdhury, Eileen Donahoe, Camille Francois, Gillian Hadfield, Eva Kailli, Safiya Noble, Navrina Singh, Katharina Borchert, Deep Ganguli, Stef Van Grieken, Abishek Gupta, Verita Harding, William Isaac, Debora Raji, Seb Krier, Kyle Miller, Russell Wald, and Daniel Zhang. Ai audit challenge.

[62] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*, 2019.

[63] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*, 2018.

[64] Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. *Advances in Neural Information Processing Systems*, 34:5656–5668, 2021.

[65] Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Are neural nets modular? inspecting functional modularity through differentiable weight masks. *arXiv preprint arXiv:2010.02066*, 2020.

[66] Tianyu Cui, Yogesh Kumar, Pekka Marttinen, and Samuel Kaski. Deconfounded representation similarity for comparison of neural networks. *arXiv preprint arXiv:2202.00095*, 2022.

[67] Allan Dafoe. Ai governance: Opportunity and theory of impact. In *Effective Altruism Forum, 17th September*, 2020.

[68] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

[69] Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. A survey of the state of explainable ai for natural language processing. *arXiv preprint arXiv:2010.00711*, 2020.

[70] Joel Dapello, Tiago Marques, Martin Schrimpf, Franziska Geiger, David Cox, and James J DiCarlo. Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations. *Advances in Neural Information Processing Systems*, 33:13073–13087, 2020.

[71] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv preprint arXiv:2006.11371*, 2020.

[72] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128, 2019.

[73] Adrien Deliège, Anthony Cioppa, and Marc Van Droogenbroeck. An effective hit-or-miss layer favoring feature interpretation as learned prototypes deformations. *arXiv preprint arXiv:1911.05588*, 2019.

[74] Jean-Stanislas Denain and Jacob Steinhardt. Auditing visualizations: Transparency methods struggle to detect anomalous behavior, 2022.

[75] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*, 2019.

[76] Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity with statistical testing. *arXiv preprint arXiv:2108.01661*, 2021.

[77] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. *Advances in Neural Information Processing Systems*, 32, 2019.

[78] Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017.

[79] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning, 2017.

[80] Keke Du, Shan Chang, Huixiang Wen, and Hao Zhang. Fighting adversarial images with interpretable gradients. In *ACM Turing Award Celebration Conference-China (ACM TURC 2021)*, pages 44–48, 2021.

[81] Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. Analyzing individual neurons in pre-trained language models. *arXiv preprint arXiv:2010.02695*, 2020.

[82] Henry Eigen and Amir Sadovnik. Topkconv: Increased adversarial robustness through deeper interpretability. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 15–22. IEEE, 2021.

[83] Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

[84] N Elhage, N Nanda, C Olsson, T Henighan, N Joseph, B Mann, A Askell, Y Bai, A Chen, T Conerly, et al. A mathematical framework for transformer circuits, 2021.

[85] Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Andy Jones, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. Softmax linear units. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/solu/index.html.

[86] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

[87] Daniel C Elton. Self-explaining ai as an alternative to interpretable ai. In *International conference on artificial general intelligence*, pages 95–106. Springer, 2020.

[88] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019.

[89] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Learning perceptually-aligned representations via adversarial robustness. *arXiv preprint arXiv:1906.00945*, 2(3):5, 2019.

[90] Christian Etmann, Sebastian Lunz, Peter Maass, and Carola-Bibiane Schönlieb. On the connection between adversarial robustness and saliency map interpretability. *arXiv preprint arXiv:1905.04172*, 2019.

[91] Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139, 2016.

[92] Ashkan Farhangi, Ning Sui, Nan Hua, Haiyan Bai, Arthur Huang, and Zhishan Guo. Protoformer: Embedding prototypes for transformers. In *Advances in Knowledge Discovery and Data Mining*, pages 447–458. Springer International Publishing, 2022.

[93] Yunzhen Feng, Runtian Zhai, Di He, Liwei Wang, and Bin Dong. Transferred discrepancy: Quantifying the difference between representations. *arXiv preprint arXiv:2007.12446*, 2020.

[94] James Fiacco, Samridhi Choudhary, and Carolyn Penstein Rosé. Deep neural model inspection and comparison via functional neuron pathways. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[95] Daniel Filan. Mechanistic transparency for machine learning.

[96] Daniel Filan, Stephen Casper, Shlomi Hod, Cody Wild, Andrew Critch, and Stuart Russell. Clusterability in neural networks. *arXiv preprint arXiv:2103.03386*, 2021.

[97] Chris Finlay and Adam M Oberman. Scaleable input gradient regularization for adversarial robustness. *arXiv preprint arXiv:1905.11468*, 2019.

[98] Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. Causal analysis of syntactic agreement mechanisms in neural language models. *arXiv preprint arXiv:2106.06087*, 2021.

[99] Hidde Fokkema, Rianne de Heide, and Tim van Erven. Attribution-based explanations that provide recourse cannot be robust. *arXiv preprint arXiv:2205.15834*, 2022.

[100] Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8730–8738, 2018.

[101] Jonathan Frankle and David Bau. Dissecting pruned neural networks. *CoRR*, abs/1907.00262, 2019.

[102] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[103] Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ö Arık, Larry S Davis, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *European Conference on Computer Vision*, pages 510–526. Springer, 2020.

[104] Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith Ranasinghe, and Hyoungshick Kim. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing*, 2021.

[105] Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586, 2021.

[106] Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, pages 7324–7338. PMLR, 2022.

[107] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness, 2018.

[108] Mor Geva, Avi Caciularu, Guy Dar, Paul Roit, Shoval Sadde, Micah Shlain, Bar Tamir, and Yoav Goldberg. Lm-debugger: An interactive tool for inspection and intervention in transformer-based language models. *arXiv preprint arXiv:2204.12130*, 2022.

[109] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*, 2022.

[110] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.

[111] Amirata Ghorbani and James Zou. Neuron shapley: Discovering the responsible neurons, 2020.

[112] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[113] Charles Godfrey, Davis Brown, Tegan Emerson, and Henry Kvinge. On the symmetries of deep learning models and their internal representations. *arXiv preprint arXiv:2205.14258*, 2022.

[114] Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. https://distill.pub/2021/multimodal-neurons.

[115] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

[116] Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763*, 2019.

[117] Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, 2015.

[118] Chris Hamblin, Talia Konkle, and George Alvarez. Pruning for interpretable, feature-preserving circuits in cnns. *arXiv preprint arXiv:2206.01627*, 2022.

[119] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12963–12971, 2021.

[120] Alexander Hartl, Maximilian Bachl, Joachim Fabini, and Tanja Zseby. Explainability and adversarial robustness for rnns. In *2020 IEEE Sixth International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 148–156. IEEE, 2020.

[121] Tiantian He, Zhibin Li, Yongshun Gong, Yazhou Yao, Xiushan Nie, and Yilong Yin. Exploring linear feature disentanglement for neural networks. *arXiv preprint arXiv:2203.11700*, 2022.

[122] Yang He, Yuhang Ding, Ping Liu, Linchao Zhu, Hanwang Zhang, and Yi Yang. Learning filter pruning criteria for deep convolutional neural networks acceleration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2009–2018, 2020.

[123] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *European conference on computer vision*, pages 3–19. Springer, 2016.

[124] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 264–279, 2018.

[125] Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. In *International Conference on Learning Representations*, 2021.

[126] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[127] Shlomi Hod, Stephen Casper, Daniel Filan, Cody Wild, Andrew Critch, and Stuart Russell. Quantifying local specialization in deep neural networks. *arXiv preprint arXiv:2110.08058*, 2021.

[128] Adrian Hoffmann, Claudio Fanconi, Rahul Rade, and Jonas Kohler. This looks like that... does it? shortcomings of latent space prototype interpretability in deep networks. *arXiv preprint arXiv:2105.02968*, 2021.

[129] Lars Holmberg. Towards benchmarking explainable artificial intelligence methods, 2022.

[130] Christopher R Hoyt and Art B Owen. Probing neural networks with t-sne, class-specific projections and a guided tour. *arXiv preprint arXiv:2107.12547*, 2021.

[131] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

[132] Evan Hubinger. Relaxed adversarial training for inner alignment.

[133] Evan Hubinger. An overview of 11 proposals for building safe advanced ai. *arXiv preprint arXiv:2012.07532*, 2020.

[134] Evan Hubinger. Automating auditing: An ambitious concrete technical research proposal, Aug 2021.

[135] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

[136] Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*, 2020.

[137] Saachi Jain, Hannah Lawrence, Ankur Moitra, and Aleksander Madry. Distilling model failures as directions in latent space, 2022.

[138] Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

[139] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. *Advances in Neural Information Processing Systems*, 33:4211–4222, 2020.

[140] Yichen Jiang and Mohit Bansal. Self-assembling modular networks for interpretable multi-hop reasoning. *arXiv preprint arXiv:1909.05803*, 2019.

[141] Amlan Jyoti, Karthik Balaji Ganesh, Manoj Gayala, Nandita Lakshmi Tunuguntla, Sandesh Kamath, and Vineeth N Balasubramanian. On the robustness of explanations of deep neural network models: A survey. *arXiv preprint arXiv:2211.04780*, 2022.

[142] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022.

[143] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models, 2020. _eprint: 2001.08361.

[144] Simran Kaur, Jeremy Cohen, and Zachary C Lipton. Are perceptually-aligned gradients a general property of robust classifiers? *arXiv preprint arXiv:1910.08640*, 2019.

[145] Been Kim, Cynthia Rudin, and Julie A Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. *Advances in neural information processing systems*, 27, 2014.

[146] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[147] Beomsu Kim, Junghoon Seo, and Taegyun Jeon. Bridging adversarial robustness and gradient interpretability. *arXiv preprint arXiv:1903.11626*, 2019.

[148] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018.

[149] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–578, 2018.

[150] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[151] Louis Kirsch, Julius Kunze, and David Barber. Modular networks: Learning to decompose neural computation. *Advances in neural information processing systems*, 31, 2018.

[152] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

[153] Arne Köhn. What's in an embedding? analyzing word embeddings through multilingual evaluation. 2015.

[154] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[155] Maya Krishnan. Against interpretability: a critical examination of the interpretability problem in machine learning. *Philosophy & Technology*, 33(3):487–502, 2020.

[156] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[157] Dmitry Krotov and John J Hopfield. Unsupervised learning by competing hidden units. *Proceedings of the National Academy of Sciences*, 116(16):7723–7731, 2019.

[158] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

[159] I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. Problems with shapley-value-based explanations as feature importance measures. In *International Conference on Machine Learning*, pages 5491–5500. PMLR, 2020.

[160] Sawan Kumar and Partha Talukdar. Nile: Natural language inference with faithful natural language explanations. *arXiv preprint arXiv:2005.12116*, 2020.

[161] C-C Jay Kuo, Min Zhang, Siyang Li, Jiali Duan, and Yueru Chen. Interpretable convolutional neural networks via feedforward design. *Journal of Visual Communication and Image Representation*, 60:346–359, 2019.

[162] Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. Qed: A framework and dataset for explanations in question answering. *arXiv preprint arXiv:2009.06354*, 2020.

[163] Richard D Lange, David S Rolnick, and Konrad P Kording. Clustering units in neural networks: upstream vs downstream information. *arXiv preprint arXiv:2203.11815*, 2022.

[164] Guillaume Leclerc, Hadi Salman, Andrew Ilyas, Sai Vemprala, Logan Engstrom, Vibhav Vineet, Kai Xiao, Pengchuan Zhang, Shibani Santurkar, Greg Yang, Ashish Kapoor, and Aleksander Madry. 3db: A framework for debugging computer vision models, 2021.

[165] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[166] Jaesong Lee, Joong-Hwi Shin, and Jun-Seok Kim. Interactive visualization and manipulation of attention-based neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 121–126, 2017.

[167] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. *arXiv preprint arXiv:2001.00689*, 2020.

[168] Michael Lepori and R Thomas McCoy. Picking bert's brain: Probing for linguistic dependencies in contextualized embeddings using representational similarity analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3637–3651, 2020.

[169] Belinda Z Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737*, 2021.

[170] Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. Probing via prompting, 2022.

[171] Oscar Li, Hao 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.

[172] Xuhong Li, Yves Grandvalet, Rémi Flamary, Nicolas Courty, and Dejing Dou. Representation transfer by optimal transport. *arXiv preprint arXiv:2007.06737*, 2020.

[173] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations? *arXiv preprint arXiv:1511.07543*, 2015.

[174] Yuchao Li, Shaohui Lin, Baochang Zhang, Jianzhuang Liu, David Doermann, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Exploiting kernel sparsity and entropy for interpretable cnn compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2800–2809, 2019.

[175] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[176] Ruofan Liang, Tianlin Li, Longfei Li, Jing Wang, and Quanshi Zhang. Knowledge consistency between neural networks and beyond. *arXiv preprint arXiv:1908.01581*, 2019.

[177] Adam Dahlgren Lindström, Suna Bensch, Johanna Björklund, and Frank Drewes. Probing multimodal embeddings for linguistic properties: the visual-semantic case. *arXiv preprint arXiv:2102.11115*, 2021.

[178] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[179] Shusen Liu, Tao Li, Zhimin Li, Vivek Srikumar, Valerio Pascucci, and Peer-Timo Bremer. Visual interrogation of attention-based models for natural language inference and machine comprehension. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2018.

[180] Yuntao Liu, Ankit Mondal, Abhishek Chakraborty, Michael Zuzak, Nina Jacobsen, Daniel Xing, and Ankur Srivastava. A survey on neural trojans. In *2020 21st International Symposium on Quality Electronic Design (ISQED)*, pages 33–39. IEEE, 2020.

[181] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

[182] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. A sober look at the unsupervised learning of disentangled representations and their evaluation. *arXiv preprint arXiv:2010.14766*, 2020.

[183] Max Losch, Mario Fritz, and Bernt Schiele. Interpretability beyond classification output: Semantic bottleneck networks. *arXiv preprint arXiv:1907.10882*, 2019.

[184] Max Losch, Mario Fritz, and Bernt Schiele. Semantic bottlenecks: Quantifying and improving inspectability of deep representations. *International Journal of Computer Vision*, 129(11):3136–3153, 2021.

[185] Adriano Lucieri, Muhammad Naseer Bajwa, Stephan Alexander Braun, Muhammad Imran Malik, Andreas Dengel, and Sheraz Ahmed. On interpretability of deep learning based skin lesion classifiers using concept activation vectors. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–10. IEEE, 2020.

[186] Adriano Lucieri, Muhammad Naseer Bajwa, Andreas Dengel, and Sheraz Ahmed. Explaining ai-based decision support systems using concept localization maps. In *International Conference on Neural Information Processing*, pages 185–193. Springer, 2020.

[187] Daniel Lundstrom, Tianjian Huang, and Meisam Razaviyayn. A rigorous study of integrated gradients method and extensions to internal neuron attributions. *arXiv preprint arXiv:2202.11912*, 2022.

[188] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pages 5058–5066, 2017.

[189] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[190] Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models. *arXiv preprint arXiv:2106.13314*, 2021.

[191] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[192] Puneet Mangla, Vedant Singh, and Vineeth N Balasubramanian. On saliency maps and adversarial robustness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 272–288. Springer, 2020.

[193] Mantas Mazieka, Dan Hendrycks, Huichen Li, Xiaojun Xu, Sidney Hough, Andy Zou, Arezoo Rajabi, Dawn Song, Radha Roovendran, Bo Li, and David Forsyth. Trojan detection challenge.

[194] David McNeely-White, Benjamin Sattelberg, Nathaniel Blanchard, and Ross Beveridge. Exploring the interchangeability of cnn embedding spaces. *arXiv preprint arXiv:2010.02323*, 2020.

[195] Johannes Mehrer, Courtney J Spoerer, Nikolaus Kriegeskorte, and Tim C Kietzmann. Individual differences among deep neural network models. *Nature communications*, 11(1):1–12, 2020.

[196] Clara Meister, Stefan Lazov, Isabelle Augenstein, and Ryan Cotterell. Is sparse attention more interpretable? *arXiv preprint arXiv:2106.01087*, 2021.

[197] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *arXiv preprint arXiv:2202.05262*, 2022.

[198] Richard Meyes, Constantin Waubert de Puiseau, Andres Posada-Moreno, and Tobias Meisen. Under the hood of neural networks: Characterizing learned representations by functional neuron populations and network ablations. *arXiv preprint arXiv:2004.01254*, 2020.

[199] Alessio Miaschi, Chiara Alzetta, Dominique Brunato, Felice Dell'Orletta, and Giulia Venturi. Probing tasks under pressure. 2021.

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

[201] Sarthak Mittal, Yoshua Bengio, and Guillaume Lajoie. Is a modular architecture enough?, 2022.

[202] Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022.

[203] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Interpretable machine learning–a brief history, state-of-the-art and challenges. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 417–431. Springer, 2020.

[204] Christoph Molnar, Gunnar König, Julia Herbinger, Timo Freiesleben, Susanne Dandl, Christian A Scholbeck, Giuseppe Casalicchio, Moritz Grosse-Wentrup, and Bernd Bischl. Pitfalls to avoid when interpreting machine learning models. 2020.

[205] Gemma E Moran, Dhanya Sridhar, Yixin Wang, and David M Blei. Identifiable variational autoencoders via sparse decoding. *arXiv preprint arXiv:2110.10804*, 2021.

[206] Ari S. Morcos, David G. T. Barrett, Neil C. Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization, 2018.

[207] Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

[208] Vincent C Müller and Nick Bostrom. Future progress in artificial intelligence: A survey of expert opinion. In *Fundamental issues of artificial intelligence*, pages 555–572. Springer, 2016.

[209] Neel Nanda. A mechanistic interpretability analysis of grokking.

[210] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29, 2016.

[211] Anh Nguyen, Jason Yosinski, Yoshua Bengio, Alexey Dosovitskiy, and Jeff Clune. Plug & play generative networks: Conditional iterative generation of images in latent space. *CoRR*, abs/1612.00005, 2016.

[212] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks, 2016.

[213] Anh Nguyen, Jason Yosinski, and Jeff Clune. Understanding neural networks via feature visualization: A survey. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 55–76. Springer, 2019.

[214] Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.15327*, 2020.

[215] Ian E Nielsen, Dimah Dera, Ghulam Rasool, Nidhal Bouaynaya, and Ravi P Ramachandran. Robust explainability: A tutorial on gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:2107.11400*, 2021.

[216] Timothy Niven and Hung-Yu Kao. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*, 2019.

[217] Adam Noack, Isaac Ahern, Dejing Dou, and Boyang Li. An empirical study on the relation between network interpretability and adversarial robustness. *SN Computer Science*, 2(1):1–13, 2021.

[218] Tuomas Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks, 2022.

[219] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. An overview of early vision in inceptionv1. *Distill*, 5(4):e00024–002, 2020.

[220] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001, 2020.

[221] Chris Olah, Nick Cammarata, Chelsea Voss, Ludwig Schubert, and Gabriel Goh. Naturally occurring equivariance in neural networks. *Distill*, 5(12):e00024–004, 2020.

[222] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[223] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

[224] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[225] OpenAI. Openai microscope, 2019.

[226] Toby Ord. *The precipice: Existential risk and the future of humanity*. Hachette Books, 2020.

[227] Guillermo Ortiz-Jiménez, Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Optimism in the face of adversity: Understanding and improving deep learning through adversarial robustness. *Proceedings of the IEEE*, 109(5):635–659, 2021.

[228] Mensah Kwabena Patrick, Adebayo Felix Adekoya, Ayidzoe Abra Mighty, and Baagyire Y Edward. Capsule networks–a survey. *Journal of King Saud University-computer and information sciences*, 34(1):1295–1310, 2022.

[229] Badri Patro, Shivansh Patel, and Vinay Namboodiri. Robust explanations for visual question answering. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1577–1586, 2020.

[230] William Peebles, John Peebles, Jun-Yan Zhu, Alexei Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *European Conference on Computer Vision*, pages 581–597. Springer, 2020.

[231] Christian S Perone, Roberto Silveira, and Thomas S Paula. Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*, 2018.

[232] Michael Petrov, Chelsea Voss, Ludwig Schubert, Nick Cammarata, Gabriel Goh, and Chris Olah. Weight banding. *Distill*, 6(4):e00024–009, 2021.

[233] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

[234] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021.

[235] Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan D Cotterell. Linear adversarial concept erasure. In *International Conference on Machine Learning*, pages 18400–18421. PMLR, 2022.

[236] Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? *arXiv preprint arXiv:2005.00719*, 2020.

[237] Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. Visualizing and measuring the geometry of bert. *Advances in Neural Information Processing Systems*, 32, 2019.

[238] Jie Ren, Mingjie Li, Qihan Ren, Huiqi Deng, and Quanshi Zhang. Towards axiomatic, hierarchical, and symbolic explanation for deep models. *arXiv preprint arXiv:2111.06206*, 2021.

[239] Mattia Rigotti, Christoph Miksovic, Ioana Giurgiu, Thomas Gschwind, and Paolo Scotton. Attention-based interpretability with concept transformers. In *International Conference on Learning Representations*, 2021.

[240] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[241] Andrew Ross, Isaac Lage, and Finale Doshi-Velez. The neural lasso: Local linear sparsity for interpretable explanations. In *Workshop on Transparent and Interpretable Machine Learning in Safety Critical Environments, 31st Conference on Neural Information Processing Systems*, volume 4, 2017.

[242] 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.

[243] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022.

[244] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[245] Stuart Russell. *Human compatible: Artificial intelligence and the problem of control*. Penguin, 2019.

[246] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[247] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. *arXiv preprint arXiv:2112.02902*, 2021.

[248] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017.

[249] Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6655–6659. IEEE, 2013.

[250] Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. Neuron-level interpretation of deep nlp models: A survey. *arXiv preprint arXiv:2108.13138*, 2021.

[251] Abdelrhman Saleh, Tovly Deutsch, Stephen Casper, Yonatan Belinkov, and Stuart Shieber. Probing neural dialog models for conversational understanding. *arXiv preprint arXiv:2006.08331*, 2020.

[252] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

[253] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021.

[254] Wojciech Samek and Klaus-Robert Müller. Towards explainable artificial intelligence. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 5–22. Springer, 2019.

[255] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. *Advances in Neural Information Processing Systems*, 32, 2019.

[256] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems*, 34, 2021.

[257] Anindya Sarkar, Anirban Sarkar, Sowrya Gali, and Vineeth N Balasubramanian. Get fooled for the right reason: Improving adversarial robustness through a teacher-guided curriculum learning approach. *arXiv preprint arXiv:2111.00295*, 2021.

[258] Johannes Schneider and Michalis Vlachos. Explaining neural networks by decoding layer activations. In *International Symposium on Intelligent Data Analysis*, pages 63–75. Springer, 2021.

[259] Ludwig Schubert, Chelsea Voss, Nick Cammarata, Gabriel Goh, and Chris Olah. High-low frequency detectors. *Distill*, 6(1):e00024–005, 2021.

[260] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557. PMLR, 2018.

[261] Sofia Serrano and Noah A Smith. Is attention interpretable? *arXiv preprint arXiv:1906.03731*, 2019.

[262] Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. Compute trends across three eras of machine learning, 2022.

[263] Jaime Sevilla, Pablo Villalobos, and Juan Felipe Cerón. \emphParameter Counts in Machine Learning, June 2021. Published: Alignment Forum (blog).

[264] Nir Shlezinger, Jay Whang, Yonina C Eldar, and Alexandros G Dimakis. Model-based deep learning. *arXiv preprint arXiv:2012.08405*, 2020.

[265] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[266] James Seale Smith, Junjiao Tian, Yen-Chang Hsu, and Zsolt Kira. A closer look at rehearsal-free continual learning. *arXiv preprint arXiv:2203.17269*, 2022.

[267] Suraj Srinivas and R Venkatesh Babu. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.

[268] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[269] Julian Stier, Gabriele Gianini, Michael Granitzer, and Konstantin Ziegler. Analysing neural network topologies: a game theoretic approach. *Procedia Computer Science*, 126:234–243, 2018.

[270] Hendrik Strobelt, Sebastian Gehrmann, Michael Behrisch, Adam Perer, Hanspeter Pfister, and Alexander M Rush. S eq 2s eq-v is: A visual debugging tool for sequence-to-sequence models. *IEEE transactions on visualization and computer graphics*, 25(1):353–363, 2018.

[271] Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. Spine: Sparse interpretable neural embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[272] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[273] Alex Tamkin, Trisha Singh, Davide Giovanardi, and Noah Goodman. Investigating transferability in pretrained language models. *arXiv preprint arXiv:2004.14975*, 2020.

[274] Shuai Tang, Wesley J Maddox, Charlie Dickens, Tom Diethe, and Andreas Damianou. Similarity of neural networks with gradients. *arXiv preprint arXiv:2003.11498*, 2020.

[275] Norman Tatro, Pin-Yu Chen, Payel Das, Igor Melnyk, Prasanna Sattigeri, and Rongjie Lai. Optimizing mode connectivity via neuron alignment. *Advances in Neural Information Processing Systems*, 33:15300–15311, 2020.

[276] Max Tegmark. *Life 3.0: Being human in the age of artificial intelligence*. Vintage, 2017.

[277] Michalis K Titsias, Jonathan Schwarz, Alexander G de G Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. *arXiv preprint arXiv:1901.11356*, 2019.

[278] Richard Tomsett, Amy Widdicombe, Tianwei Xing, Supriyo Chakraborty, Simon Julier, Prudhvi Gurram, Raghuveer Rao, and Mani Srivastava. Why the failure? how adversarial examples can provide insights for interpretable machine learning. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 838–845. IEEE, 2018.

[279] Joe Townsend, Theodoros Kasioumis, and Hiroya Inakoshi. Eric: extracting relations inferred from convolutions. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[280] Theodoros Tsiligkaridis and Jay Roberts. Second order optimization for adversarial robustness and interpretability. *arXiv preprint arXiv:2009.04923*, 2020.

[281] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[282] Sunil Vadera and Salem Ameen. Methods for pruning deep neural networks. *arXiv preprint arXiv:2011.00241*, 2020.

[283] Marco Valentino, Ian Pratt-Hartmann, and André Freitas. Do natural language explanations represent valid logical arguments? verifying entailment in explainable nli gold standards. *arXiv preprint arXiv:2105.01974*, 2021.

[284] Shikhar Vashishth, Shyam Upadhyay, Gaurav Singh Tomar, and Manaal Faruqui. Attention interpretability across nlp tasks. *arXiv preprint arXiv:1909.11218*, 2019.

[285] Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically interpretable reinforcement learning. In *International Conference on Machine Learning*, pages 5045–5054. PMLR, 2018.

[286] Jesse Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019.

[287] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in Neural Information Processing Systems*, 33:12388–12401, 2020.

[288] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.

[289] Chelsea Voss, Nick Cammarata, Gabriel Goh, Michael Petrov, Ludwig Schubert, Ben Egan, Swee Kiat Lim, and Chris Olah. Visualizing weights. *Distill*, 6(2):e00024–007, 2021.

[290] Chelsea Voss, Gabriel Goh, Nick Cammarata, Michael Petrov, Ludwig Schubert, and Chris Olah. Branch specialization. *Distill*, 6(4):e00024–008, 2021.

[291] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE, 2019.

[292] Jie Wang, Ghulam Mubashar Hassan, and Naveed Akhtar. A survey of neural trojan attacks and defenses in deep learning. *arXiv preprint arXiv:2202.07183*, 2022.

[293] Liwei Wang, Lunjia Hu, Jiayuan Gu, Zhiqiang Hu, Yue Wu, Kun He, and John Hopcroft. Towards understanding learning representations: To what extent do different neural networks learn the same representation. *Advances in neural information processing systems*, 31, 2018.

[294] Yulong Wang, Hang Su, Bo Zhang, and Xiaolin Hu. Interpret neural networks by identifying critical data routing paths. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8906–8914, 2018.

[295] Yulong Wang, Xiaolu Zhang, Xiaolin Hu, Bo Zhang, and Hang Su. Dynamic network pruning with interpretable layerwise channel selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6299–6306, 2020.

[296] Chihiro Watanabe. Interpreting layered neural networks via hierarchical modular representation. In *International Conference on Neural Information Processing*, pages 376–388. Springer, 2019.

[297] Chihiro Watanabe, Kaoru Hiramatsu, and Kunio Kashino. Modular representation of layered neural networks. *Neural Networks*, 97:62–73, 2018.

[298] Chihiro Watanabe, Kaoru Hiramatsu, and Kunio Kashino. Understanding community structure in layered neural networks. *Neurocomputing*, 367:84–102, 2019.

[299] John Wentworth. Testing the natural abstraction hypothesis: Project intro.

[300] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*, 2019.

[301] Olivia Wiles, Isabela Albuquerque, and Sven Gowal. Discovering bugs in vision models using off-the-shelf image generation and captioning, 2022.

[302] Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In *International Conference on Machine Learning*, pages 11205–11216. PMLR, 2021.

[303] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33:15173–15184, 2020.

[304] Mike Wu, Michael Hughes, Sonali Parbhoo, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[305] Kaidi Xu, Sijia Liu, Gaoyuan Zhang, Mengshu Sun, Pu Zhao, Quanfu Fan, Chuang Gan, and Xue Lin. Interpreting adversarial examples by activation promotion and suppression, 2019.

[306] Yujun Yan, Kevin Swersky, Danai Koutra, Parthasarathy Ranganathan, and Milad Hashemi. Neural execution engines: Learning to execute subroutines. *Advances in Neural Information Processing Systems*, 33:17298–17308, 2020.

[307] Kaixuan Yao, Feilong Cao, Yee Leung, and Jiye Liang. Deep neural network compression through interpretability-based filter pruning. *Pattern Recognition*, 119:108056, 2021.

[308] Seul-Ki Yeom, Philipp Seegerer, Sebastian Lapuschkin, Alexander Binder, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Pruning by explaining: A novel criterion for deep neural network pruning. *Pattern Recognition*, 115:107899, 2021.

[309] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

[310] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022.

[311] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017.

[312] Mengmi Zhang, Rohil Badkundri, Morgan B Talbot, Rushikesh Zawar, and Gabriel Kreiman. Hypothesis-driven online video stream learning with augmented memory. *arXiv e-prints*, pages arXiv–2104, 2021.

[313] Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[314] Quanshi Zhang, Xin Wang, Ruiming Cao, Ying Nian Wu, Feng Shi, and Song-Chun Zhu. Explanatory graphs for cnns. *arXiv preprint arXiv:1812.07997*, 2018.

[315] Quanshi Zhang, Xin Wang, Ruiming Cao, Ying Nian Wu, Feng Shi, and Song-Chun Zhu. Extraction of an explanatory graph to interpret a cnn. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3863–3877, 2020.

[316] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8827–8836, 2018.

[317] Yujia Zhang, Kuangyan Song, Yiming Sun, Sarah Tan, and Madeleine Udell. " why should you trust my explanation?" understanding uncertainty in lime explanations. *arXiv preprint arXiv:1904.12991*, 2019.

[318] Mengjie Zhao, Tao Lin, Fei Mi, Martin Jaggi, and Hinrich Schütze. Masking as an efficient alternative to finetuning for pretrained language models. *arXiv preprint arXiv:2004.12406*, 2020.

[319] Xinyan Zhao and VG Vydiswaran. Lirex: Augmenting language inference with relevant explanation. *arXiv preprint arXiv:2012.09157*, 2020.

[320] Haizhong Zheng, Earlence Fernandes, and Atul Prakash. Analyzing the interpretability robustness of self-explaining models. *arXiv preprint arXiv:1905.12429*, 2019.

[321] Songzhu Zheng, Yikai Zhang, Hubert Wagner, Mayank Goswami, and Chao Chen. Topological detection of trojaned neural networks. *Advances in Neural Information Processing Systems*, 34:17258–17272, 2021.

[322] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.

[323] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.

[324] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Revisiting the importance of individual units in cnns via ablation. *arXiv preprint arXiv:1806.02891*, 2018.

[325] Yilun Zhou, Marco Tulio Ribeiro, and Julie Shah. Exsum: From local explanations to model understanding. *arXiv preprint arXiv:2205.00130*, 2022.

[326] Daniel M Ziegler, Seraphina Nix, Lawrence Chan, Tim Bauman, Peter Schmidt-Nielsen, Tao Lin, Adam Scherlis, Noa Nabeshima, Ben Weinstein-Raun, Daniel de Haas, et al. Adversarial training for high-stakes reliability. *arXiv preprint arXiv:2205.01663*, 2022.