Interpretable by Design: Learning Predictors by Composing Interpretable Queries

Aditya Chattopadhyay, Stewart Slocum, Benjamin D. Haeffele, René Vidal, *Fellow, IEEE*, and Donald Geman, *Life Senior Member, IEEE*

Abstract—There is a growing concern about typically opaque decision-making with high-performance machine learning algorithms. Providing an explanation of the reasoning process in domain-specific terms can be crucial for adoption in risk-sensitive domains such as healthcare. We argue that machine learning algorithms should be interpretable by design and that the language in which these interpretations are expressed should be domain- and task-dependent. Consequently, we base our model's prediction on a family of user-defined and task-specific binary functions of the data, each having a clear interpretation to the end-user. We then minimize the expected number of queries needed for accurate prediction on any given input. As the solution is generally intractable, following prior work, we choose the queries sequentially based on information gain. However, in contrast to previous work, we need not assume the queries are conditionally independent. Instead, we leverage a stochastic generative model (VAE) and an MCMC algorithm (Unadjusted Langevin) to select the most informative query about the input based on previous query-answers. This enables the online determination of a query chain of whatever depth is required to resolve prediction ambiguities. Finally, experiments on vision and NLP tasks demonstrate the efficacy of our approach and its superiority over post-hoc explanations.

Index Terms—Explainable AI, interpretable ML, computer vision, generative models, information theory

1 Introduction

In recent years, interpreting large machine learning models has emerged as a major priority, particularly for transparency in making decisions or predictions that impact human lives [1], [2], [3]. In such domains, understanding *how* a prediction is made may be as important as achieving high predictive accuracy. For example, medical regulatory agencies have recently emphasized the need for computational algorithms used in diagnosing, predicting a prognosis, or suggesting treatment for a disease, to explain why a particular decision was made [4], [5].

On the other hand, it is widely believed that there exists a fundamental trade-off in machine learning between interpretability and predictive performance [6], [7], [8], [9], [10]. Simple models like decision trees and linear classifiers are often regarded as *interpretable*[1] but at the cost of potentially reduced accuracy compared with larger *black box* models such as deep neural networks. As a result, considerable effort has been given to developing methods that provide *post-hoc* explanations of black box model predictions, i.e., given a prediction from a (fixed) model provide additional annotation or elaboration to explain how the prediction was made. As a concrete example, for image classification problems, one common family of post-hoc explanation methods produces attribution maps which seek to estimate the regions of the image that are *most important* for prediction. This is typically approached by attempting to capture the effect or sensitivity of perturbations to the input (or intermediate features) on the model output [11], [12], [13], [14], [15], [16], [17], [18]. However, post-hoc analysis has been critiqued for a variety of issues [2], [19], [20], [21], [22], [23] (see also Section 2) and often fails to provide explanations in terms of concepts that are intuitive or interpretable for humans [24].

This naturally leads to the question of what an *ideal* explanation of a model prediction would entail; however, this is potentially highly *task-dependent* both in terms of the task itself as well as what the user seeks to obtain from an explanation. For instance, a model for image classification is often considered interpretable if its decision can be explained in terms of patterns occurring in salient parts of the image [25] (e.g., the image is a car because there are wheels, a windshield, and doors), whereas in a medical task explanations in terms of causality and mechanism could be desired (e.g., the patient's chest pain and shortness of breath is likely not a pulmonary embolism because the blood D-dimer level is low, suggesting thrombosis is unlikely). Note that some words or patterns may be *domain-dependent* and therefore not interpretable to non-experts, and hence what is interpretable ultimately depends on the end user, namely the person who is trying to understand or deconstruct the decision made by the algorithm [26].

1. Although later in the paper we will discuss situations in which even these simple models need not be interpretable.

• *The authors are with the Mathematical Institute for Data Science, Center for Imaging Science, Johns Hopkins University, Baltimore, MD 21218 USA. E-mail: {achatto1, sslocum3, bhaeffele, rvidal, geman}@jhu.edu.*

Manuscript received 12 May 2022; revised 7 October 2022; accepted 12 November 2022. Date of publication 28 November 2022; date of current version 5 May 2023.
This work was supported in part by Army Research Office under the Multidisciplinary University Research Initiative under Grant W911NF-17-1-0304 and in part by the NSF under Grant 2031985.
(Corresponding author: Aditya Chattopadhyay.)
Recommended for acceptance by A. van den Hengel.
This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2022.3225162, provided by the authors.
Digital Object Identifier no. 10.1109/TPAMI.2022.3225162

In addition to this *task-dependent* nature of model interpretation, there are several other desirable intuitive aspects of interpretable decisions that one can observe. The first is that meaningful interpretations are often *compositional* and can be constructed and explained from a set of *elementary units* [27]. For instance, words, parts of an image, or domain-specific concepts [28], [29], [30] could all be a suitable basis to form an explanation of a model's prediction depending on the task. Moveover, the basic principle that simple and *concise* explanations are preferred (i.e., Occam's razor) suggests that interpretablity is enhanced when an explanation can be composed from the smallest number of these elementary units as possible. Finally, we would like this explanation to be *sufficient* for describing model predictions, meaning that there should be no external variables affecting the prediction that are not accounted for by the explanation.

Inspired by these desirable properties, we propose a framework for learning predictors that are *interpretable by design*. The proposed framework is based on composing a subset of user-defined *concepts*, i.e., functions of the input data which we refer to as *queries*, to arrive at the final prediction. Possible choices for the set of queries Q based on the style of interpretation that is desired include:

1) *Salient image parts:*For vision problems, if one is interested in explanations in terms of salient image regions then this can be easily accomplished in our framework by defining the query set to be a collection of small patches (or even single pixels) within an image. This can be thought of as a generalization of the pixel-wise explanations generated by attribution maps.
2) *Concept-based explanations:*In domains such as medical diagnosis or species identification, the user might prefer explanations in terms of concepts identified by the community to be relevant for the task. For instance, a "Crow" is determined by the shape of the beak, color of the feathers, etc. In our framework, by simply choosing a query for each such concept, the user can easily obtain concept-based explanations (see Fig. 1b).
3) *Visual scene interpretation:*In visual scene understanding, one seeks a rich semantic description of a scene by accumulating the answers to queries about the existence of objects and relationships, perhaps generating a scene graph [31]. One can design a query set Q by instantiating these queries with trained classifiers. The answers to chosen queries in this context would serve as a semantic interpretation of the scene.
4) *Deep neuron-based explanations:* The above three examples are query sets based on domain knowledge. Recent techniques [30], [32], [33] have shown the ability of different neurons in a trained deep network to act as concept detectors. These are learnt from data by solving auxiliary tasks without any explicit supervisory signal. One could then design a Q in which each query corresponds to the activation level of a specific concept neuron. Such a query set will be useful for tasks in which it is difficult to specify interpretable functions/queries beforehand.

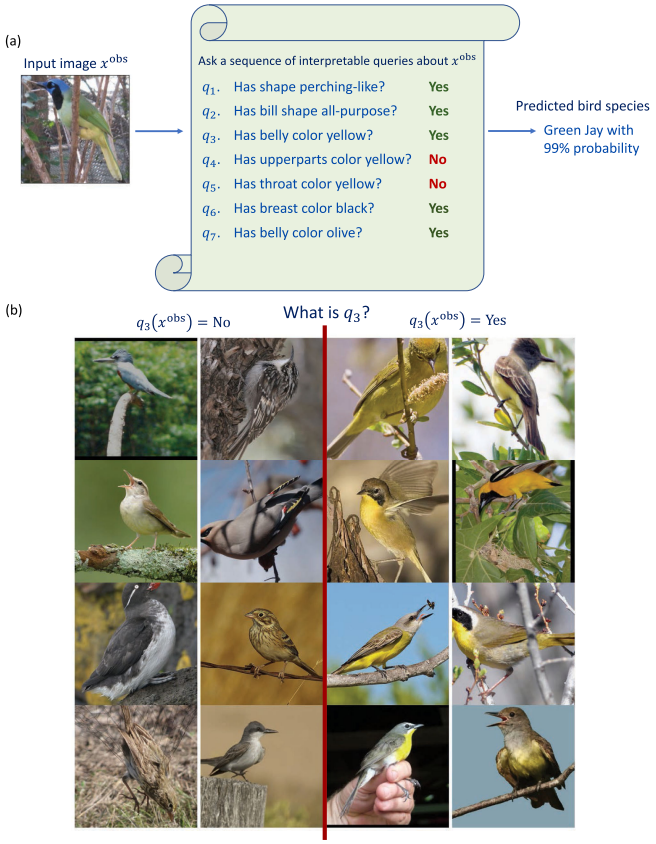

Fig. 1. **(a) An illustration of our proposed learning framework.** The prediction of a bird species is explained through a short sequence of interpretable queries, $(q_1, q_2, \ldots, q_7)$, derived from a user-defined query set of domain-specific attribute for birds. **(b) Interpretable queries.** Each query in this case corresponds to a well-defined bird attribute. For instance, q_3 asks "Does the bird have belly color yellow?". We visualize some example images which evaluate to "Yes" and observe that all of them correspond to birds with a yellow belly. Similarly, all images which evaluate to "No" corresponds to birds which do not have a yellow belly.

Given a user-specified set of queries Q, our framework makes its prediction by selecting a short sequence of queries such that the sequence of query-answer pairs provides a complete explanation for the prediction. More specifically, the selection of queries is done by first learning a generative model for the joint distribution of queries and output labels and then using this model to select the "most informative" queries for a given input. The final prediction is made using the Maximum A Posteriori (MAP) estimate of the output given these query-answer pairs. Fig. 1a gives an illustration of our proposed framework, where the task is to predict the bird species in an image and the queries are based on color, texture and shape attributes of birds. We argue that the sequence of query-answer pairs provides a meaningful explanation to the user that captures the subjective nature of interpretability depending on the task at hand, and that is, by construction, compositional, concise and sufficient.

At first glance, one might think that classical decision trees [34], [35] based on Q could also produce interpretable decisions by design. However, the classical approach to determining decision tree branching rules based on the empirical distribution of the data is prone to over-fitting due to data fragmentation. Whereas random forests [36], [37] are often much more competitive than classical decision trees in accuracy [38], [39], [40], they sacrifice

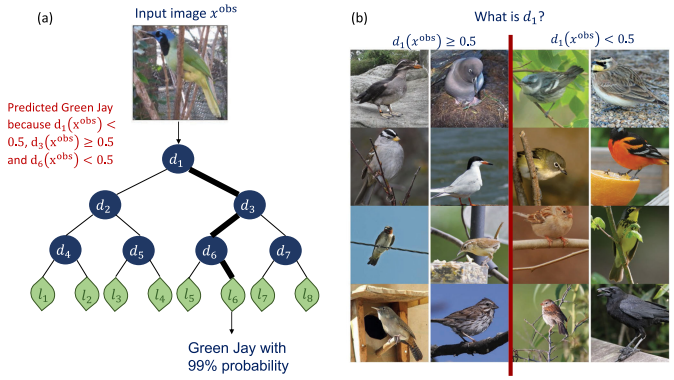

Fig. 2. **The interpretability of an explanation depends on how interpretable the queries are. (a)** An illustration of a Deep Neural decision tree [41] trained on the CUB-2011 dataset of bird images. The bold path denotes the trajectory the input image x^{obs} takes through the tree. Each d_i corresponds to an internal node of the tree and is a black-box function/query learnt from data. Each l_i denotes a leaf and computes the final classification for x^{obs}. The prediction can be explained as a conjunction of internal node functions, but is it really interpretable? **(b)** Example images that get routed to the left sub-tree ($d_1 \geq 0.5$) and right sub-tree ($d_1 < 0.5$) of the root node. Notice that the interpretation of d_1 is not clear from these examples. Compare this to Fig. 1 where the semantics of each query is unambiguous to the end-user.

interpretability, the very property we want to hardwire into our decision algorithm. Similarly, the accuracy of a single tree can be improved by using deep networks to learn queries directly from data, as in Neural Decision Trees (NDTs) [41]. However, the opaqueness of the interpretation of these learnt queries makes the explanation of the final output, in terms of logical operations on the queries at the internal nodes, unintelligible. Fig. 2 illustrates this with an example.

In this paper we make the following contributions;

- We propose a novel framework for prediction that is *interpretable by design*. We allow the end-user to specify a set Q of queries about input X and formulate learning as the problem of selecting a minimal set of queries from Q whose answers are sufficient for predicting output Y. We formulate this query selection problem as an optimization problem over strategies that minimize the number of queries needed on average to predict Y from X. A prediction for Y is then made based on the selected query-answer pairs, which provide an explanation for the prediction that is by construction interpretable. The set of selected query-answer pairs can be viewed as a *code* for the input. However, a major difference between our framework and coding theory is that, due to the constraint of interpretability, Q is a vanishingly small collection of the functions of X, whereas coding theory typically considers Q to be all possible binary functions of X.
- Since computing the exact solution to our optimization problem is computationally challenging, we propose to greedily select a minimal set of queries by using the *Information Pursuit* (IP) algorithm [42]. IP sequentially selects queries in order of maximum *information gain* until enough evidence is gathered from the query-answer pairs to predict Y. This sequence of query-answer pairs serves as the *explanation* for predicting Y from X. To ameliorate the computational challenge of computing information gain for high-dimensional input and query spaces, prior work [42] had assumed that query answers were conditionally independent given Y, an assumption that is largely inadequate for most prediction tasks we encounter in practice. In this paper, we propose a latent variable graphical model for the joint distribution of queries and outputs, $p(Q(X), Y)$, and learn the required distributions using Variational Autoencoders (VAEs). We then use the Unadjusted Langevin Algorithm (ULA) to generate samples required to carry out IP. This gives us a tractable algorithm for any task and query set. To the best of our knowledge, ours is the first implementation of IP that uses deep generative models and does *not* assume that query answers are conditionally independent given Y.
- Finally, we demonstrate the utility of our framework on various vision and NLP tasks. In binary image classification using MNIST, Fashion-MNIST & KMNIST, and bird species identification using CUB-200, we observe that IP finds succinct explanations which are highly predictive of the class label. We also show, across various datasets, that the explanations generated by our method are shorter and more predictive of the class label than state-of-the-art post-hoc explanation methods like Integrated Gradients and DeepSHAP.

2 Related Work

Methods for interpretable deep learning can be separated into those that seek to explain existing models (post-hoc methods) and those that build models that are interpretable by design. Because they do not negatively impact performance and are convenient to use, post-hoc explanations have been the more popular approach, and include a great diversity of methods.

Saliency maps estimate the contribution of each feature through first-order derivatives [11], [12], [16], [17], [43]. Linear perturbation-based methods like LIME [44] train a linear model to locally approximate a deep network around a particular input, and use the coefficients of this model to estimate the contribution of each feature to the prediction. Another popular set of methods use game-theoretic Shapley values as attribution scores, estimating feature contributions by generating predictions on randomly sampled subsets of the input [45]. We provide quantitative comparisons between IP and these methods in Section 5.1.2. Recently, there has been interest in concept-based analogues of these methods that leverage similar approaches to measure the sensitivity of a prediction to high-level, human-friendly concepts as opposed to raw features [46], [47], [48].

Despite certain advantages, what all the above post-hoc methods have in common is that they come with little guarantee that the explanations they produce actually reflect how the model works [2]. Indeed, several recent studies [18], [19], [20], [21], [22] call into question the veracity of these explanations towards the trained model. Adebayo et al. [19] show that several popular attribution methods act

similar to edge detectors and are insensitive to the parameters of the model they attempt to explain! Yang et al.[20] find that these methods often produce false-positive explanations, assigning importance to features that are irrelevant to the prediction of the model. It is also possible to adversarially manipulate post-hoc explanations to hide any spurious biases the trained model might have picked up from data [23].

Interpretability by Design. These issues have motivated recent work on deep learning models which are *interpretable by design*, i.e., constrained to produce explanations that are faithful to the underlying model, albeit with varying conceptions of "faithfulness". Several of these models are constructed so they behave similarly to or can be well-approximated by a classically interpretable model, such as a linear classifier [49], [50] or a decision tree [51]. This allows for an approximately faithful explanation in raw feature space. In a similar vein, Pillai & Pirsiavash [52] fix a post-hoc explanation method (e.g., Grad-CAM [16]), and regularize a model to generate consistent explanations with the chosen post-hoc method. However, our method does not just behave *like* a fully interpretable model or generate *approximately* faithful explanations, but rather it produces explanations that are guaranteed to be faithful and fully explain a given prediction.

Another approach to building interpretable models by design is to generate explanations in terms of high-level, interpretable concepts rather than in raw feature space, often by applying a linear classifier to a final latent space of concepts [25], [49], [53]. However these concepts are learned from data, and may not align with the key concepts identified by the user. For example, Prototypical Part Networks [25] take standard convolutional architectures and insert a "prototype layer" before the final linear layer, learning a fixed number of visual concepts that are used to represent the input. This allows the network to explain a prediction in terms of these "prototype" concepts. Since these prototypes are learned embeddings, there is no guarantee that their interpretation will coincide with the user's requirements. Furthermore, these explanations may require a very large number of concepts, while in contrast, we seek minimal-length explanations to preserve interpretability.

Attention-based models are another popular family of models that are sometimes considered interpretable by design [54], [55]. However, attention is only a small part of the overall computation and can be easily manipulated to hide model biases [56]. Moreover, the attention coefficients are not necessarily a sufficient statistic for the model prediction.

Perhaps most similar to our work are Concept Bottleneck Networks [24], which first predict an intermediate set of human-specified concepts c and then use c to predict the final class label. Nevertheless, the learnt mapping from concepts to labels is still a black-box. To remedy this, the authors suggest using a linear layer for this mapping but this can be limiting since linearity is often an unrealistic assumption [27]. In contrast, our framework makes no linearity assumptions about the final classifier and the classification is explainable as a sequence of interpretable query-answer pairs obtained about the input (see Fig. 1a).

Neural Networks and Decision Trees. Unlike the above methods, which can be thought of as deep interpretable linear classifiers, our method can be described as a deep decision tree that branches on responses to an interpretable query set. Spanning decades, there has been a variety of work building decision trees from trained neural networks [29], [57], [58], [59] and using neural networks within nodes of decision trees [41], [60], [61], [62]. Our work differs from these in three important aspects. First, rather than allowing arbitrary splits, we branch on responses to an interpretable query set. Second, instead of using empirical estimates of information gains based on training data (which inevitably encounter data-fragmentation [63] and hence overfitting), or using heuristics like agglomerative clustering on deep representations [29], we calculate information gain from a generative model, leading to strong generalization. Third, for a given input, say x^{obs}, we use a generative model to compute the queries along the branch traversed by x^{obs} in an online manner. The entire tree is never constructed. This allows for much very deep terminal nodes when necessary to resolve ambiguities in prediction. As an example, for the task of topic classification using the HuffPost dataset (Section 5.0.3), our framework asks about 199 queries (on average) before identifying the topic. Such large depths are impossible in standard decision trees due to memory limitations.

Information Bottleneck and Minimal Sufficient Statistics. The problem of finding minimal-length, task-sufficient codes is not new. For example, the *information bottleneck* method [64] seeks a minimum-length encoding for X that is (approximately) sufficient to solve task Y. Our concept of description length differs in that we constrain the code to consist of interpretable query functions rather than *all functions* of the input, as in the information bottleneck and classical information theory. Indeed, arbitrary subsets of the input space (e.g., images) are overwhelmingly *not* interpretable to humans.

Sequential Active Testing and Hard Attention. The *information pursuit* (IP) algorithm we use was introduced in [42] under the name "active testing," which sequentially observes parts of an input (rather than the whole input at once), using mutual information to determine "where to look next," which is calculated online. Sequentially guiding the selection of partial observations has also been independently explored in Bayesian experimental design [65]. Subsequent works in these two areas include many ingredients of our approach (e.g., generative models [31], [66] and MCMC algorithms [67]). Of particular interest is the work of Branson et al. [68] which used the CUB dataset to identify bird species by sequentially asking pose and attribute queries to a human user. They employ IP to generate the query sequence based on answers provided by the user, much like our experiments in Section 5.0.2. However, for the sake of tractability, all the above works assume that query answers are independent conditioned on Y. We do not. Rather, to the best of our knowledge, ours is the first implementation of the IP algorithm that uses deep generative models and only assumes that queries are independent given Y *and some latent variable* Z. This greatly improves performance, as we show in Section 5.

The strategy of inference through sequential observation of portions of the input has been recently re-branded in the deep learning community as *Hard Attention* [69], [70], [71]. However, high variance in gradient estimates and scalability issues have prevented widespread adoption. In the future, we wish to explore how our work could inform more principled and better-performing reward functions for Hard Attention models.

Visual Question Answering. Although it may appear that our work is also related to the Visual Question Answering (VQA) literature [72], [73], [74], [75], [76], [77], we note that our work addresses a very different problem. VQA focuses on training deep networks for *answering* a large set of questions about a visual scene. In contrast, our framework is concerned with *selecting* a small number of queries to ask about a given image to solve a task, say classification. As we move on to more complex tasks, an interesting avenue for future work would involve using VQA systems to supply answers to the queries used in our framework. However, this would require significantly more complex generative models that the ones considered here.

3 Learning Interpretable Predictors by Composing Queries via Information Pursuit

Let X and Y be the input data and the corresponding output/hypothesis, both random variables assuming values in $\mathcal{X}$ and $\mathcal{Y}$ respectively. In supervised learning, we seek to infer Y from X using a finite set of samples drawn from the joint distribution $p_{XY}(x, y)$.[2] As motivated in Section 1, useful explanations for prediction should be *task-dependent*, *compositional*, *concise* and *sufficient*. We capture such properties through a suitably rich set Q of binary functions $q(x)$, or *queries*, whose answers $\{q(x)\}_{q\in Q}$ collectively determine the task Y. More precisely, a query set Q is *sufficient* for Y if

$$p(y \mid x) = p(y \mid \{x' \in \mathcal{X} : q(x') = q(x)\ \forall q \in Q\}). \tag{1}$$

In other words, Q is sufficient for Y if whenever two inputs x and x' have identical answers for all queries in Q, their corresponding posteriors are equal, i.e., $p(y \mid x) = p(y \mid x')$.

Given a fixed query set Q, how do we compose queries into meaningful representations that are predictive of Y? We answer this by first formally defining an explanation strategy π and then formulating the task of composing queries as an optimization problem.

Explanation Strategies Based on Composing Queries. An *explanation strategy*, or just *strategy*, is a function, $\pi : K^* \rightarrow Q$, where K^* is the set of all finite-length sequences generated using elements from the set $K = \{(q, q(x)) \mid q \in Q, x \in \mathcal{X}\}$ of query-answer pairs. We require that Q contains a special query, q_{STOP}, which signals the strategy to stop asking queries and output $expl^{\pi}_{Q}(x)$, the set of query-answer pairs asked before q_{STOP}. More formally, a strategy π is recursively defined as follows; given input sample x^{obs}

1) $q_1 = \pi(\emptyset)$. The first query is independent of x^{obs}.
2) $q_{k+1} = \pi(\{q_i, q_i(x^{\text{obs}})\}_{1:k})$. All subsequent queries depend on the query-answer pairs observed so far for x^{obs}.
3) If $q_{L+1} = q_{STOP}$ terminate, and return

$$expl^{\pi}_{Q}(x^{\text{obs}}) := \{q_i, q_i(x^{\text{obs}})\}_{1:L}. \tag{2}$$

Notice that each q_i depends on x^{obs}, but we drop this dependency in the notation for brevity. We call the number of pre-STOP queries for a particular x^{obs} as the explanations' description length and denote it by $t^{\pi}(x^{\text{obs}}) := |expl^{\pi}_{Q}(x^{\text{obs}})|$. Computing a strategy on x^{obs} is thus akin to traversing down the branch of a decision tree dictated by x^{obs}. Each internal node encountered along this branch computes the query proposed by the strategy based on the path (query-answer pairs) observed so far.

Notice also that we restrict out attention to *sequential* strategies so that the resulting explanations satisfy the property of being *prefix-free*.[3] This means that explanations generated for predictions made on an input signal x_1 cannot be a sub-part for explanations generated for predictions on a different input signal x_2; otherwise, the explanation procedure is ambiguous because a terminal node carrying one label could be an internal node of a continuation leading to a different label. Sequential strategies generate prefix-free explanations by design. For non-sequential strategies, which are just functions mapping an input X to a set of queries in Q, it is not clear how to effectively encode the constraint of generating prefix-free explanations.

Concise and Approximately Sufficient Strategies. In machine learning, we are often interested in solving a task *approximately* rather than *exactly*. Let Q be sufficient for Y, choose a distance-like metric d on probability distributions and let $\epsilon > 0$. We propose the following optimization problem to efficiently compose queries for prediction

$$\begin{aligned} &\min_{\pi} \mathbb{E}_X\Big[|expl^{\pi}_{Q}(X)|\Big] =: H^{\epsilon}_{Q}(X;Y) \\ &\text{s.t. } \mathbb{E}_X[d\Big(p(Y \mid X), p(Y \mid expl^{\pi}_{Q}(X)\Big)] \leq \epsilon \ \ (\epsilon - Sufficiency), \end{aligned} \tag{3}$$

where the minimum is taken over all strategies π. The solution π^* to (3) provides a criterion for an optimal strategy for the task of inferring Y approximately from X. The *minimal expected description length* objective, $H^{\epsilon}_{Q}(X;Y)$, ensures the conciseness of the explanations, while the constraint ensures approximate sufficiency of the explanation. The "metric" d on distributions could be KL-divergence, total variation, Wasserstein distance, etc. The hyper-parameter ϵ controls how approximate the explanations are. The posterior $p(y \mid expl^{\pi}_{Q}(x^{\text{obs}}))$ should be interpreted as the conditional probability of y given the event

$$[x^{\text{obs}}]_{\pi,Q} := \{x \in \mathcal{X} \mid expl^{\pi}_{Q}(x) = expl^{\pi}_{Q}(x^{\text{obs}})\}. \tag{4}$$

$expl^{\pi}_{Q}(X)$ can also be interpreted as a random variable which maps input X to its equivalence class $[X]_{\pi,Q}$.

2. We denote random variables by capital letters and their realizations with small letters.

3. The term prefix-free comes from the literature on instantaneous codes in information theory.

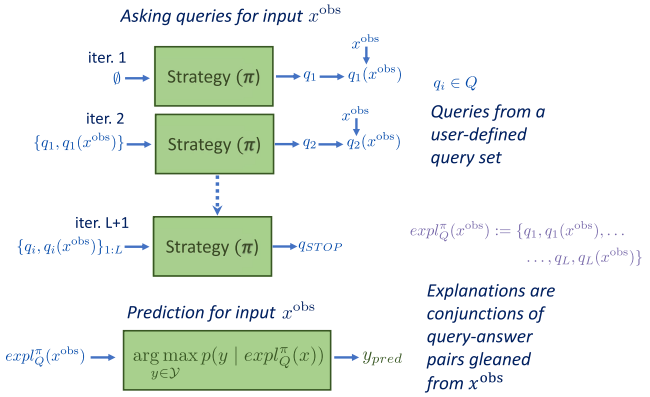

Fig. 3. Schematic view of the overall framework for quantifying explanations for predicting y from x^{obs}. For details see Section 3.

The final prediction/inference for the input x^{obs} is then taken to be the usual MAP estimator, namely

$$y_{pred} = \arg\max_{y \in \mathcal{Y}} p(y \,|\, expl_Q^\pi(x^{\text{obs}})). \tag{5}$$

The sequence of query-answers streams obtained by π on x^{obs} serves as the explanation for y_{pred}. One could also monitor the posterior over the labels Y evolving as successive queries get asked to gain more insight into the strategy's decision-making process. Fig. 3 illustrates the overall framework in detail.

Information Pursuit: A Greedy Approximation. Unfortunately, solving (3) is known to be NP-Complete and hence generally intractable [78]. As an approximate solution to (3) we propose to use a greedy algorithm called Information Pursuit (IP). IP was introduced by Geman & Jedynak in 1996 [42] as a model-based, online construction of a single but deep branch. The IP strategy, that is, $\pi = \text{IP}$, is recursively defined as follows:

$$\begin{aligned} q_1 &= \text{IP}(\emptyset) = \arg\max_{q \in Q} I(q(X); Y) \\ q_{k+1} &= \text{IP}(\{q_i, q_i(x^{\text{obs}})\}_{1:k}) = \arg\max_{q \in Q} I(q(X); Y \,|\, S_k^{\text{IP}}(x^{\text{obs}})) \end{aligned} \tag{6}$$

where I denotes mutual information and $S_k^{\text{IP}}(x^{\text{obs}})$ corresponds to the event $\{x \in \mathcal{X} \,|\, \{q_i, q_i(x^{\text{obs}})\}_{1:k} = \{q_i, q_i(x)\}_{1:k}\}$. Ties in choosing q_{k+1} are broken arbitrarily if the maximum is not unique.

The algorithm stops when there are no more informative queries left in Q, that is, it satisfies the following condition

$$\begin{aligned} q_{L+1} = q_{STOP} \quad \text{if} \quad & \max_{q \in Q} I(q(X); Y \,|\, S_m^{\text{IP}}(x^{\text{obs}})) \le \epsilon \\ & \forall m \in \{L, L+1, \ldots, L+T\}, \end{aligned} \tag{7}$$

where hyper-parameter $T > 0$ is chosen via cross-validation. This termination criteria corresponds to taking the distance-like metric d in (3) as the KL-divergence between the two distributions. Further details about the relation between this termination criteria and the ϵ-Sufficiency constraint in (3) are provided in Appendix A.3, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3225162. For tasks in which Y is a function of X, a common scenario in many supervised learning problems, we use a simpler alternative

$$\begin{aligned} q_{L+1} = q_{STOP} \quad \text{if} \quad & \arg\max_{y \in \mathcal{Y}} p(y \,|\, S_m^{\text{IP}}(x^{\text{obs}})) \ge 1 - \epsilon \\ & \forall m \in \{L, L+1, \ldots, L+T\}. \end{aligned} \tag{8}$$

The key distinction between the information gain criteria used in standard decision tree induction and IP is that the former uses the empirical distributions to compute (6) while the latter is based on generative models (as we will see in Section 4). The use of generative models guards against data fragmentation [63] and thus allows for asking longer sequences of queries without grossly over-fitting.

How Does IP *Compare to the Optimal Strategy* $\boldsymbol{\pi^*}$*?* We begin by characterizing the constraint in (3) in terms of mutual information, the quantity that drives IP.

Proposition 1. *Let $S_k^\pi(X)$ be a random variable where any realization $S_k^\pi(x^{\text{obs}})$, $x^{\text{obs}} \in \mathcal{X}$, denotes the event*

$$S_k^\pi(x^{\text{obs}}) := \{x' \in \mathcal{X} \,|\, \{q_i, q_i(x^{\text{obs}})\}_{1:k} = \{q_i, q_i(x')\}_{1:k}\},$$

where q_i is the i^{th} query selected by π for input x^{obs}. Here we use the convention that $S_0^\pi(X) = \Omega$ (the entire sample space) and $S_l^\pi(X) = S_{t^\pi(X)}^\pi(X) \;\; \forall l > t^\pi(X)$. If Q is finite[4] and d is taken to be the KL-divergence, then objective (3) can be rewritten as

$$\begin{aligned} & H_Q^\epsilon(X; Y) := \min_\pi \mathbb{E}_X \Big[|expl_Q^\pi(X)| \Big] \\ & \text{s.t.} \sum_{k=1}^{\tau^\pi} I(Y; S_k^\pi(X) \,|\, S_{k-1}^\pi(X)) \ge I(X; Y) - \epsilon, \end{aligned} \tag{9}$$

where $\tau^\pi = \max\{t^\pi(x) : x \in \mathcal{X}\}$ and $t^\pi(X)$ is defined as the number of queries selected by π for input X until q_{STOP}.

See Appendix A.1, available in the online supplemental material, for a detailed proof. The objective in (9) can be alternatively stated as

$$\begin{aligned} & \max_\pi \sum_{k=1}^{\tau^\pi} I(Y; S_k^\pi(X) \,|\, S_{k-1}^\pi(X)) \\ & \text{s.t.} \; \mathbb{E}_X \Big[|expl_Q^\pi(X)| \Big] \le \gamma, \end{aligned} \tag{10}$$

where $\gamma > 0$ is a user-defined hyper-parameter. From (10) it is clear that the optimal strategy π^* would ask a sequence of queries about X that would maximize the cumulative sum of the mutual information each additional query provides about Y, conditioned on the history of query-answers observed so far, subject to a constraint on the average number of queries that can be asked. As stated before, solving for π^* is infeasible but a greedy approximation that makes locally optimal choices is much more amenable.

Suppose that one has been given the answers to k queries about a given input, the locally optimal choice would then be to ask the most informative query about Y conditioned on the history of these k query-answers observed. This greedy choice at each stage gives rise to the IP strategy.

4. The assumption of Q being a finite set is benign. Many interested applications can be addressed with a finite Q as we show in our experiments.

Obtaining approximation guarantees for IP is still an open problem; however in the special case where Q is taken to be the set of all possible binary functions of X, it is possible to show that IP *asks at most 1 query more* than π^* on average. More formally, we have the following result, whose proof can be found in Appendix A.2, available in the online supplemental material.

Proposition 2. *Let Y be discrete. Let $\tilde{H}_Q(X;Y)$ be the expected description length obtained by the* IP *strategy. If $H(Y|X) = 0$ and Q is the set of all possible binary functions of X such that $H(q(X)\,|\,Y) = 0\ \forall q \in Q$, then*

$$H(Y) \leq \tilde{H}_Q(X;Y) \leq H(Y) + 1 \tag{11}$$

Having posed the problem of finding explanations as an optimization problem and proposed a greedy approximation to solving it, in the next section we propose a tractable implementation of IP based on deep generative models.

4 Information Pursuit Using Variational Autoencoders and Unadjusted Langevin

IP requires probabilistic models relating query-answers and data to compute the required mutual information terms in (6). Specifically, computing q_{k+1} in (6) (for any iteration number k) requires computing the mutual information between $q(X)$ and Y given the history $S_k^{\text{IP}}(x^{\text{obs}})$ till time k. As histories become longer, we quickly run out of samples in our dataset which belong to the event $S_k^{\text{IP}}(x^{\text{obs}})$. As a result, non-parametric sample-based methods to estimate mutual information (such as [79]) would be impractical. In this section, we propose a model-based approach to address this challenge for a general supervised learning task and query set Q. In Section 5 we adapt this model to the specific cases where Q is taken to be image patches or task-based concepts.

Information Pursuit Generative Model. To make learning tractable, we introduce latent variables Z to account for all the dependencies between different query-answers, and we posit the following factorization of $Q(X), Y, Z$

$$p_{Q(X)ZY}(Q(x), z, y) = \prod_{q\in Q} p_{q(X)\,|\,ZY}(q(x)\,|\,z, y)p_Y(y)p_Z(z), \tag{12}$$

where $Q(X) = \{q(X) : q \in Q\}$, and z and $q(x)$ denote realizations of Z and $q(X)$ respectively. In other words, we assume that the query-answers are conditionally independent given the label y and a latent vector z. The independence assumption in (12) shows up ubiquitously in many machine learning applications, such as the following.

1) $\boldsymbol{q(X)}$ *as object presence indicators evaluated at non-overlapping windows:* Let Q be a set of non-overlapping windows in the image X with $q(X)$ being a random variable indicating the presence of an object at the q^{th} location. The correlation between the qs is entirely due to latent image generating factors Z, such as lighting, camera position, scene layout, and texture along with the scene description signal Y.
2) $\boldsymbol{q(X)}$ *as snippets of speech utterances:* A common assumption in speech recognition tasks is that the audio frame features ($q(X)$) are conditionally independent given latent phonemes Z (which is often modeled as a Hidden Markov Model).

The latent space Z is often a lower-dimensional space compared to the original high-dimensional X. We learn Z from data in an unsupervised manner using variational inference. Specifically, we parameterize the distributions $\{p_\omega(q(x)\,|\,z, y)\ \forall q \in Q\}$ with a *Decoder Network* with shared weights ω. These weights are learned using stochastic Variational Bayes [80] by introducing an approximate posterior distribution $p'_\phi(z\,|\,y, Q(x))$ parameterized by another neural network with weights ϕ called the *Encoder Network* and priors $p_Y(y)$ and $p_Z(z)$. More specifically, the parameters ϕ and ω are learned by maximizing the Evidence Lower BOund (ELBO) objective. Appendix A.7, available in the online supplemental material, gives more details on this optimization procedure. The learned Decoder Network $p_{\omega^*}(q(x)\,|\,z, y)$ is then used as a plug-in estimate for the true distribution $p_{q(X)\,|\,ZY}(q(x)\,|\,z, y)$, which is in turn used to estimate (12).

Implementing **IP** *Using the Generative Model.* Once the Decoder Network has been learned using variational inference, the first query $q_1 = \text{IP}(\emptyset)$ is the one that maximizes the mutual information with Y as per (6). The mutual information term for any query q is completely determined by $p(q(x), y)$, which is obtained by numerically marginalizing the nuisances Z from (12) using Monte Carlo integration. In particular, we carry out the following computation $\forall q \in Q$

$$\begin{aligned} p_{q(X)Y}(q(x), y) &= \int_z p_{Q(X)ZY}(Q(x), z, y)dz \\ &= \int_z p_{q(X)\,|\,ZY}(q(x)\,|\,z, y)p_Y(y)p_Z(z)dz \\ &\approx \frac{1}{N}\sum_{i=1}^{N} p_{\omega^*}(q(x)\,|\,y, z^{(i)})p_Y(y) \\ &=: \tilde{p}(q(x), y)). \end{aligned} \tag{13}$$

In the last approximation, $p_{\omega^*}(q(x)\,|\,y, z^{(i)})$ is the distribution obtained using the trained decoder network. N is the number of i.i.d. samples drawn and $z^i \sim p_Z(z)$. We then estimate mutual information numerically via the following formula

$$I(Y; q(X)) = \sum_{q(x),y} \tilde{p}(q(x), y) \log \frac{\tilde{p}(q(x), y)}{\tilde{p}(q(x))\tilde{p}(y)}. \tag{14}$$

The computation of subsequent queries q_{k+1} requires the mutual information conditioned on observed history $S_k^{\text{IP}}(x^{\text{obs}})$, which can be calculated from the distribution

$$\begin{aligned} &p(q(x), y\,|\,S_k^{\text{IP}}(x^{\text{obs}})) \\ &= \int p(q(x), z, y\,|\,S_k^{\text{IP}}(x^{\text{obs}}))dz \\ &= \int p(q(x)\,|\,z, y, S_k^{\text{IP}}(x^{\text{obs}}))p(z\,|\,y, S_k^{\text{IP}}(x^{\text{obs}}))p(y\,|\,S_k^{\text{IP}}(x^{\text{obs}}))dz \\ &= \int p(q(x)\,|\,z, y)p(z\,|\,y, S_k^{\text{IP}}(x^{\text{obs}}))p(y\,|\,S_k^{\text{IP}}(x^{\text{obs}}))dz. \end{aligned} \tag{15}$$

The first equality is an application of the law of total probability. The last equality appeals to the assumption that $\{q(X), q \in Q\}$ are conditionally independent given Y, Z (12).

To estimate the right-hand side of (15) via Monte Carlo integration, one needs to sample $z^i \sim p(z \mid y, S_k^{\text{IP}}(x^{\text{obs}}))$ and compute

$$p(q(x), y \mid S_k^{\text{IP}}(x^{\text{obs}})) \approx \tilde{p}(q(x), y \mid S_k^{\text{IP}}(x^{\text{obs}})) := \frac{1}{N}\sum_{i=1}^{N} p_{\omega^*}(q(x)|z^{(i)}, y)p(y|S^{\text{IP}_{k(x^{\text{obs}}))}}, \quad (16)$$

where the term $p(y \mid S_k^{\text{IP}}(x^{\text{obs}}))$ is estimated recursively via the Bayes' theorem. This computation is as follows:

$$\begin{aligned} p(y \mid S_k^{\text{IP}}(x)) &\propto p(y, S_k^{\text{IP}}(x)) \\ &= p(q_k(x), y, S_{k-1}^{\text{IP}}(x)) \\ &\propto p(q_k(x) \mid y, S_{k-1}^{\text{IP}}(x))p(y \mid S_{k-1}^{\text{IP}}(x)) \end{aligned} \quad (17)$$

$S_0^{\text{IP}}(x) = \emptyset$ (since no evidence via queries has been gathered from x yet) and so $p(y \mid S_0^{\text{IP}}(x)) = p_Y(y)$. The posterior $p(y \mid S_k^{\text{IP}}(x))$ is obtained by normalizing the last equation in (17) such that $\sum_y p(y|S_k^{\text{IP}}(x)) = 1$. This recursive updating of the posterior is similar to the posterior updates used in Bayesian sequential filtering [81]. The term $p(q_k(x) \mid y, S_{k-1}^{\text{IP}}(x))$ is estimated using (16).

Having estimated $p(q(x), y \mid S_k^{\text{IP}}(x^{\text{obs}}))$, we then numerically compute the mutual information between query-answer $q(X)$ and Y given history for every $q \in Q$ via the formula

$$I(Y; q(X) \mid S_k^{\text{IP}}(x^{\text{obs}})) = \sum_{q(x),y} \tilde{p}(q(x), y|S_k^{\text{IP}}(x^{\text{obs}}))\log \frac{\tilde{p}(q(x), y \mid S_k^{\text{IP}}(x^{\text{obs}}))}{\tilde{p}(q(x)|S_k^{\text{IP}}(x^{\text{obs}}))\tilde{p}(y|S_k^{\text{IP}}(x^{\text{obs}}))}. \quad (18)$$

Estimating $\boldsymbol{p(z \mid y, S_k^{\text{IP}}(x^{\text{obs}}))}$ with the Unadjusted Langevin Algorithm. Next we describe how to sample from this posterior $p(z \mid y, S_k^{\text{IP}}(x^{\text{obs}}))$ using the Unadjusted Langevin Algorithm (ULA). ULA is an iterative algorithm used to approximately sample from any distribution with a density known only up to a normalizing factor. It has been successfully applied to many high-dimensional Bayesian inference problems [82], [83], [84]. Given an initialization $z^{(0)}$, ULA proceeds by

$$z^{(i+1)} = z^{(i)} + \eta \nabla U(z^{(i)}) + \sqrt{2\eta}\zeta^{(i+1)}. \quad (19)$$

Here $(\zeta^{(i)})_{i\geq 1} \sim \mathcal{N}(0, I)$ and η is the step-size. Asymptotically, the chain $(z^{(i)})_{i\geq 1}$ converges to a stationary distribution that is "approximately" equal to a measure with density $\propto e^{U(z)}$ [85].

For IP, we need samples from $p(z \mid y, S_k^{\text{IP}}(x^{\text{obs}})$. This is achieved by initializing $z^{(0)}$ using the last iterate of the ULA chain used to simulate $p(z \mid y, S_{k-1}^{\text{IP}}(x^{\text{obs}})$.[5] We then run ULA for N iterations by recursively applying (19) with

$$U(z) := \log p(z, S_k^{\text{IP}}(x^{\text{obs}})|y) = \log p(S_k^{\text{IP}}(x^{\text{obs}})|z, y)p(z)p(y).$$

The number of steps N is chosen to be sufficiently large to ensure the ULA chain converges "approximately" to the desired $z \sim p(z \mid y, S_k^{\text{IP}}(x^{\text{obs}}))$. We use the trained decoder network $\prod_{i=1}^{k} p_\omega(q_i(x) \mid z, y)$, with q_i being the i^{th} query asked by IP for input x, as a proxy for $p(S_k^{\text{IP}}(x^{\text{obs}})) \mid z, y)$. We then obtain stochastic approximations of (15) by time averaging the iterates

$$p\big(q(x), y|S_k^{\text{IP}}(x^{\text{obs}})\big) \approx \frac{1}{N}\sum_{i=1}^{N} p_\omega\Big(q(x)|z^{(i)}, y\Big)p\big(y|S_k^{\text{IP}}(x^{\text{obs}})\big), \quad (20)$$

where $(z^{(i)})_{1:N}$ are the iterates obtained using the ULA chain whose stationary distribution is "approximately" $p(z \mid y, S_k^{\text{IP}}(x^{\text{obs}}))$.

Algorithmic Complexity for IP. For any given input x, the per-iteration cost of the IP algorithm is $\mathcal{O}(N + |Q|m)$[6], where $|Q|$ is the total number of queries, N is the number of ULA iterations, and m is cardinality of the product sample space $q(X) \times Y$. For simplicity we assume that the output hypothesis Y and query-answers $q(X)$ are finite-valued and also that the number of values query answers can take is the same. However, our framework can handle more general cases. See Appendix A.6 for more details, available in the online supplemental material.

5 Experiments

In this section, we empirically evaluate the effectiveness of our method. We begin by analyzing the explanations provided by IP for classifying individual input data, in terms of words, symbols, or patterns (the queries). We find in each case that IP discovers concise explanations which are amenable to human interpretation. We then perform quantitative comparisons which show that (i) IP explanations are more faithful to the underlying model than existing attribution methods; and (ii) the predictive accuracy of our method using a given query set is competitive with black-box models trained on features provided by the same set.

5.0.1 Binary Image Classification With Patch Queries

Task and Query Set. We start with the simple task of binary image classification. We consider three popular datasets – MNIST [86], Fashion-MNIST[87] and KMNIST [88]. We choose a threshold for binarizing these datasets since they are originally grayscale. We choose the query set Q as the set of all $w \times w$ overlapping patch locations in the image. The answer $q(X)$ for any $q \in Q$ is the w^2 pixel intensities observed at the patch indexed by location q. This choice of Q reflects the user's desire to know which parts of the input image are most informative for a particular prediction, a common practice for explainability in vision tasks [25]. We conduct experiments for multiple values of w and conclude that $w = 3$ provides a good trade-off between the required number of queries and the interpretability of each query. Note that when $w > 1$ the factorization in (12) that we use to model $p(Q(x), y, z)$ and compute mutual information no longer holds as the overlapping queries $q(X)$ are now *causally related* (and therefore dependent even when conditioned on Z, making them unable to be modeled by a VAE). So instead of training a VAE to directly model the query set

5. $z^{(0)} \sim \mathcal{N}(0, I)$ for the first iteration of IP.

6. In this computation we have assumed, for simplicity, a unit cost for any operation that was computed in a batch concurrently on a GPU.

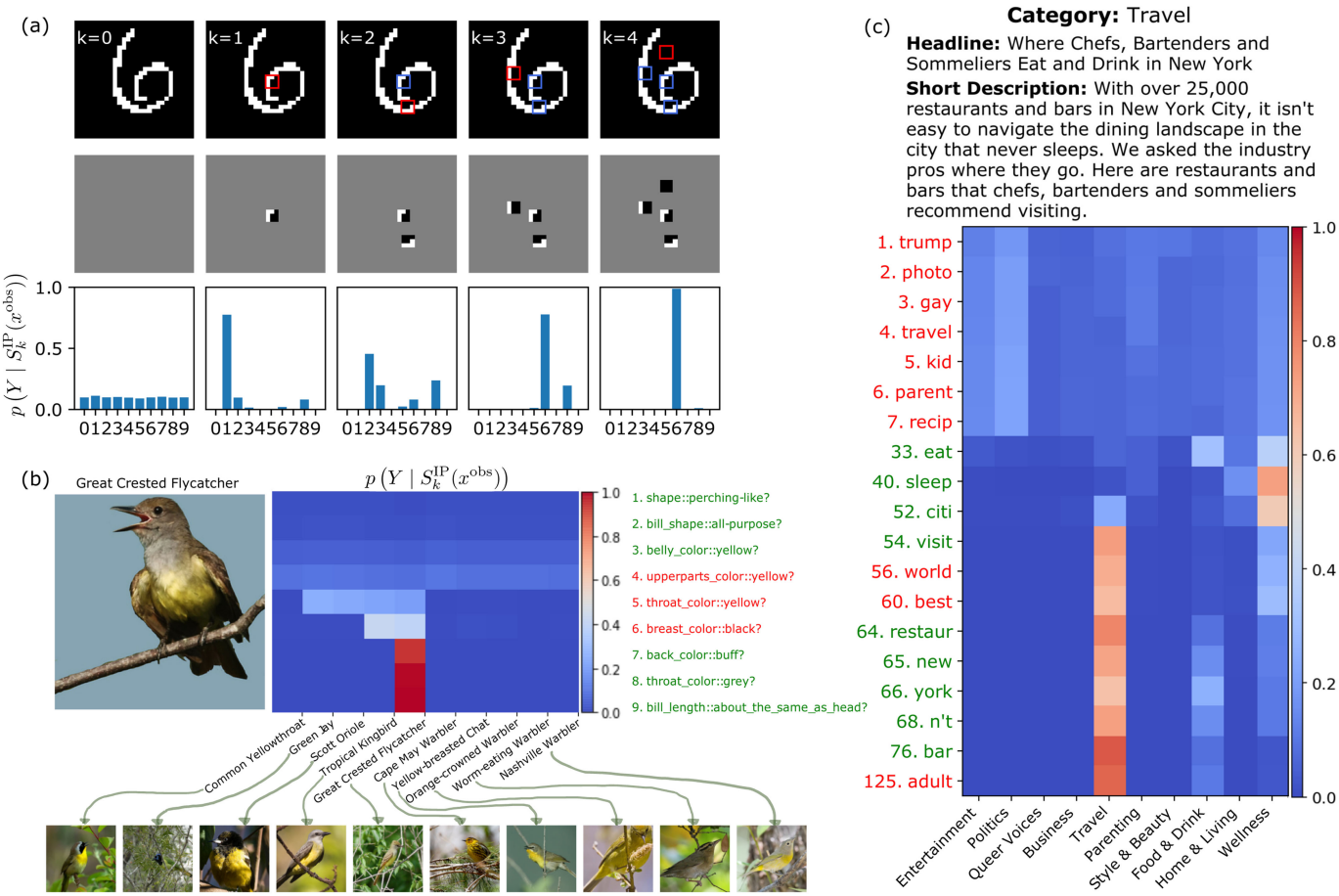

Fig. 4. **(a) IP on MNIST.** The top row displays the test image with red boxes denoting the current queried patch and blue boxes denoting previous patches. The second row shows the revealed portion of the image that IP gets to use at each query. The final row shows the model's estimated posteriors at each query, beginning at a nearly uniform prior before converging on the true digit "6" after 4 queries. **(b) IP on CUB Bird Species Classification.** On the left we show the input image and on the right we have a heatmap of the estimated class probabilities at each iteration. We only show the top 10 most probable classes out of the 200. To the right, we display the queries asked at each iteration, with red indicating a "no" response and green a "yes" response. **(c) IP on HuffPost News.** We show the input news item and a heatmap depicting the evolution of topic probabilities as IP asks queries and gathers answers. Words colored in red are absent from the sentence while words in green are present. For our visualization, we compute the KL divergence between each successive posterior and plot only the top 20 queries that led to the greatest change in posterior class probabilities.

$p(Q(x) \mid y, z)$, we train a VAE to model the pixel distribution $p(x \mid y, z)$, and then compute the probability distribution over the patch query $p(q(x) \mid z, y)$ as the product of the probabilities of all pixels in that patch.[7]

IP in Action. Fig. 4a illustrates the decision-making process of IP using 3×3 patch queries on an image x^{obs} of a 6 from the MNIST test set. The first query is near the center of the image; recall from (6) that this choice is independent of the particular input image and represents the patch whose pixel intensities have maximum mutual information with Y (the class label). The updated posterior, $p(Y \mid S_1^{\mathrm{IP}}(x^{\mathrm{obs}}))$, concentrates most of its mass on the digit "1", perhaps because most of the other digits do not commonly have a vertical piece of stroke at the center patch. However, the next query (about three pixels below the center patch) reveals a horizontal stroke and the posterior mass over the labels immediately shifts to $\{2, 3, 6, 8\}$. The next two queries are well-suited to discerning between these four possibilities and we see that after asking 4 questions, IP is more than 90% confident that the image is a 6. Such rich explanations in terms of querying informative patches based on what is observed so far and seeing how the belief $p(Y \mid S_k^{\mathrm{IP}}(x^{\mathrm{obs}}))$ of the model evolves over time is missing from post-hoc attribution methods which output static importance scores for every pixel towards the black-box model's final prediction.

Explanation Length Versus Task Complexity. Fig. 6 shows that IP requires an average of 5.2, 12.9 and 14.5 queries of size 3×3 to predict the label with 99% confidence ($\epsilon = 0.01$ in (8)) on MNIST, KMNIST and FashionMNIST, respectively. This reflects the intuition that more complex tasks require longer explanations. For reference, state-of-the-art deep networks on these datasets obtain test accuracies in order MNIST $\geq$ KMNIST $\geq$ FashionMNIST (see last row in Table 1).

Effect of Patch Size on Interpretability. We also run IP on MNIST with patch sizes of 1×1 (single pixels), 2×2, 3×3, and 4×4. We observed that IP terminates at 99% confidence after 21.1, 9.6, 5.2, and 4.6 queries on average, respectively. While this suggests that larger patches lead to shorter explanations, we note that explanations with larger patches use more pixels (e.g., on MNIST, IP uses 21.1 pixels on average for 1×1 patches and 54.7 pixels on average for 4×4

7. Since the patches overlap in our query set, when computing the conditional probability of a patch query given history we only consider the probability of the pixels in the patch that have not yet been observed in our history.

TABLE 1
Classification Accuracy of Our Model (Information Pursuit) Relative to Baselines on Different Test Sets

Model	MNIST	KMNIST	Fashion	CUB	HuffPost
Information Pursuit	96.78%	91.02%	85.60%	76.73%	71.21%
Decision Tree[34]	90.23%	78.00%	80.80%	68.80%	63.00%
MAP using Q	99.05%	94.25%	87.56%	76.80%	71.72%
Black-box using Q	99.15%	95.10%	88.43%	76.30%	71.48%
Black-box	99.83% [93]	98.83% [88]	96.70% [87]	82.70% [24]	86.45%[8]

See 5.1.1 for details on each model.

patches). That being said, very small patch queries are hard to interpret (see Fig. 5) and very large patch queries are also hard to interpret since each patch contains many image features. Overall, we found that 3×3 patches represented the right trade-off between interpretability in terms of edge patterns and minimality of the explanations. Specifically, single pixels are not very interpretable to humans but the explanations generated are more efficient in terms of *number of pixels needed to predict the label*. On the other extreme, using the entire image as a query is not interesting from an interpretability point of view since it does not help us understand which parts of the image are salient for prediction. We refer the reader to Appendix B.3.1, available in the online supplemental material, for additional patch size examples and quantitative analysis.

5.0.2 Concept-Based Queries

Task and Query Set. What if the end-user is interested in a more semantic explanation of the decision process in terms of high-level concepts? This can be easily incorporated into our framework by choosing an appropriate query set Q. As an example we consider the challenging task of bird species classification on the Caltech-UCSD Birds-200-2011 (CUB) dataset [89]. The dataset contains images of 200 different species of birds. Each image is annotated with 312 binary attributes representing high-level concepts, such as the colour and shape of the beak, wings, and head. Unfortunately, these attribute annotations are very noisy. We follow [24] in deciding attribute labels by majority voting. For example, if more than 50% of images in a class have black wings, then we set all images in that class to have black wings. We construct Q by choosing a query for asking the presence/absence of each of these 312 binary attributes. Unfortunately, attribute annotations are not available at test time. To remedy this, we train a CNN (see [24] for details) to answer each query using the training set annotations, which is then used to answer queries at test time. Subsequently, we learn a VAE to model the joint distribution of query-answers supplied by this CNN (instead of the ground truth annotations) and Y, so our generative model can account for any estimation errors incurred by the CNN. Finally, we carry out IP as explained in Section 4.

8. We fine-tuned a Bert Large Uncased Transformer model [92] with the last layer replaced with a linear one. See Appendix B.2.3, available in the online supplemental material, for details.

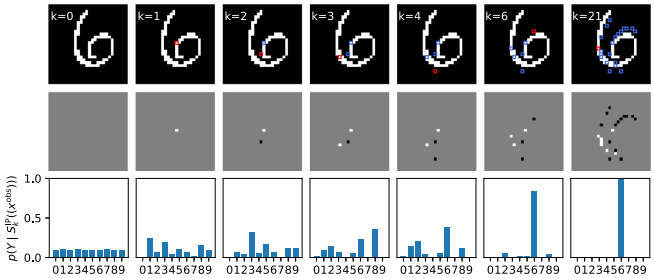

Fig. 5. IP with 1×1 patches on MNIST. Through the first 6 iterations, IP asks queries in the same center vertical region as in Fig. 4a (which uses 3×3 queries), outlining the distinctive loop in the bottom of the "6". However, reaching 99% confidence requires a total of 21 1×1 queries as opposed to just 4 3×3 ones. For conciseness, we show only the 6 queries that led to the greatest KL divergence between successive posterior class probabilities.

IP in Action. Consider the image of a *Great Crested Flycatcher* in Fig. 4b. IP proceeds by asking most informative queries about various bird attributes progressively making the posterior over the species labels more and more peaked. After 5 queries, IP has gathered that the image is of a bird that has a perching-like shape, all-purpose beak and yellow belly, but does not have a yellow throat nor yellow upperparts. This results in a posterior concentrated on just 4 species that exhibit these characteristics. IP then proceeds to discount *Green Jay* and *Scott Oriole* which have black breasts with query 6. Likewise, *Tropical Kingbirds* have grayish back and is segregated from *Great Crested Flycatchers* which have buff-coloured backs with query 7. Finally after 9 queries, IP is 99% confident about the current class. Such concept-based explanations are more accessible to non-experts, especially on fine-grained classification datasets, which typically require domain expertise. On average IP takes 14.7 queries to classify a given bird image with $\epsilon = 0.007$ as the stopping criteria (See (7)).

5.0.3 Word-Based Queries

Task and Query Set. Our framework can also be successfully applied to other domains like NLP. As an example we consider the task of topic identification from newspaper extended headlines (headline + short description field) using the the Huffington Post News Category Dataset [90]. We adopt a simple query set that consists of binary queries probing the existence of words in the extended headline. The words are chosen from a pre-defined vocabulary obtained by stemming all words in the HuffPost dataset and choosing the top-1,000 according to their tf-idf scores [91]. We process the dataset to merge redundant categories (such as *Style & Beauty* and *Beauty & Style*), remove semantically ambiguous, HuffPost-specific categories (e.g., *Impact* or *Fifty*) and remove categories with few samples, arriving at 10 final categories (see Appendix B.1, available in the online supplemental material).

IP in Action. Fig. 4c shows an example run of IP on the HuffPost dataset. Note that positive responses to queries are very sparse, since each extended headline only contains 8.6 words on average out of the 1,000 in the vocabulary. As a result, IP asks 125 queries before termination. As discussed in Section 2, such long decision paths would be impossible in decision trees due to data fragmentation and memory limitations. For clarity of presentation we only

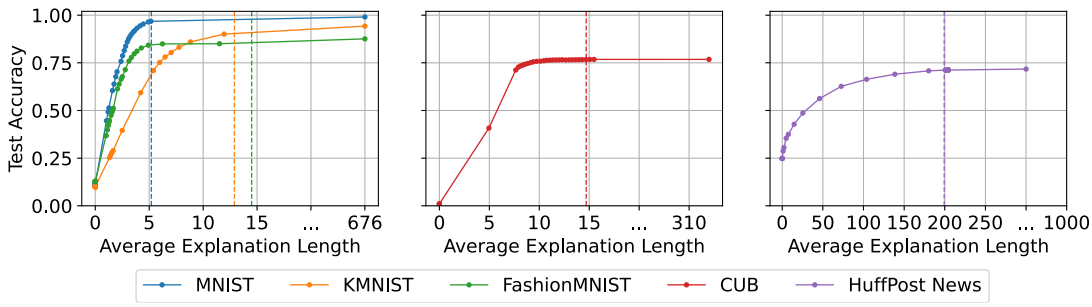

Fig. 6. Trade-off between predictive performance and explanation length Different points along the curves correspond to different values of ϵ as the stopping criteria (7) is varied. The colored dotted vertical line in each plot indicates the avg. explanation length v/s test accuracy at the ϵ value used as the stopping criteria for reporting results for the IP strategy in this work. For each plot, the x-axis ranges from 0 to the size of the query set, $|Q|$, chosen for that task.

show the 20 queries with the greatest impact on the estimated posterior (as measured by KL-divergence from previous posterior). Upon reaching the first positive query "eat", the probability mass concentrates on the categories *Food & Drink* and *Wellness* with little mass on *Travel*. However, as the queries about the existence of "citi", "visit", "york", and "bar" in the extended headline come back positive, the model becomes more and more confident that "Travel" is the correct class. IP requires about 199.3 queries on average to predict the topic of the extended headline with $\epsilon = 10^{-3}$ as the stopping criteria (See (7)). Additional details on the HuffPost query set are in Appendix B.1, available in the online supplemental material.

Further examples of IP performing inference on all tasks can be found in Appendix B.3, available in the online supplemental material.

5.1 Quantitative Evaluation

5.1.1 Classification Accuracy

We compare the classification accuracy of our model's prediction based on the query-answers gathered by IP until termination with several other baseline models. For each of the models considered, we first give a brief description and then comment on their performance with respect to IP. All the results are summarized in Table 1.

DECISION TREE refers to standard classification trees learnt using the popular CART algorithm [34]. In the Introduction, we mentioned that classical decision trees learnt using Q to supply the node splitting functions will be intepretable by construction but are not competitive with state-of-the-art methods. This is illustrated in our results in Table 1. Across all datasets, IP obtains superior performance since it is based on an underlying generative model (VAE) and only computes the branch of the tree traversed by the input data in an online manner, thus it is not shackled by data fragmentation and memory limitations.

MAP USING Q refers to the Maximum A Posteriori estimate obtained using the posterior distribution over the labels given the answers to all the queries in Q (for a given input). Recall, IP asks queries until the stopping criteria is reached (Equations (7) & (8)). Naturally, there is a trade-off between the length of the explanations and the predictive performance observed. If we ask all the queries then the resulting explanations of length $|Q|$ might be too long to be desirable. The results for IP reported in Table 1 use different dataset-specific stopping criteria according to the elbow in their respective accuracy versus explanation length curves (see Fig. 6). On the binary image datasets, (MNIST, KMNIST, and FashionMNIST) IP obtains an accuracy within 3% of the best achievable upon seeing all the query-answers with only about 2% of the total queries in Q. Similarly for the CUB and Huffpost datasets, IP achieves about the same accuracy as MAP USING Q but asks less than 5% and 20% of total possible queries respectively.

BLACK-BOX USING Q refers to the best performing deep network model we get by training on features supplied by evaluating all $q \in Q$ on input data from the various training datasets. For the binary image datasets, this is just a 4-layer CNN with ReLU activations. For CUB we use the results reported by the sequential model in [24]. For HuffPost, we found a single hidden layer with ReLU non-linearity give the best performance. Further architectural and training details are in Appendix B.2, available in the online supplemental material. In Table 1 we show that across all datasets, the predictive performance obtained by MAP USING Q is on par with the best performance we obtained using black-box expressive non-interpretable networks BLACK-BOX USING Q. Thus, our generative models, which form the backbone for IP, are competitive with state-of-the-art prediction methods.

BLACK-BOX refers to the best performing black-box model on these datasets in terms of classification accuracy as reported in literature; to the best of our knowledge. In Table 1, we see a performance gap in each dataset when compared with MAP USING Q which uses an interpretable query set. This is expected since explainability can be viewed as an additional constraint on learning. For example, on FashionMNIST we see an almost 8.5% relative fall in accuracy due to binarization. This is because it is harder to decipher between some classes like shirts and pullovers at the binary level. On the other hand, binary patches are easily interpretable as edges, foregrounds and backgrounds. Similarly, there is a relative drop of accuracy of about 17% for the HuffPost dataset since our queries regarding the existence of different words ignore their arrangement in sentences. Thus we lose crucial contextual information used by state-of-the-art transformer models [92]. Ideally, we would like query sets to be easily interpretable, lead to short explanations and be sufficient to solve the task. Finding such query sets is nontrivial and will be explored in future work.

5.1.2 Comparison to Current Attribution Methods

At first glance, it might seem that using attribution methods/saliency maps can provide the same insights as to

TABLE 2
MAP Accuracy of Explanations Generated by Information Pursuit (IP) Versus Other Attribution Methods

Explanation Method	MNIST	KMNIST	Fashion-MNIST
Information Pursuit	**96.78%**	**91.02%**	85.60%
IG	78.48%	84.87%	78.49%
IG (absolute)	70.39%	84.72%	64.95%
DeepSHAP	87.98%	88.90%	**88.36%**
DeepSHAP (absolute)	84.80%	84.56%	84.35%

IP explanations (in almost all cases) achieve a higher classification accuracy than explanations of the same length generated using baseline attribution methods. The (absolute) method refers to explanations generated using absolute values of the attribution map scores. On MNIST and KMNIST, IP explanations achieve a 10% *and* 2.38% *relative improvement respectively over the best performing baseline method. On FashionMNIST, IP explanations are second best with a relative decrease of about* 3.12% *from the best performing baseline.*

which parts of the image or more generally which queries in Q were most influential in a decision made by a black-box model trained on input features supplied by all the query-answers. However, the unreliability of these methods in being faithful to the model they try to explain brings their utility into question [19], [20], [22]. We conjecture that this is because current attribution methods are not designed to generate explanations that are sufficient statistics of the model's prediction. We illustrate this with a simple experiment using our binary image classification datasets.

For each input image x, we compute the corresponding attribution map $e(x)$ for the model's predicted class using two popular attribution methods, Integrated gradients (IG) [94] and DeepSHAP [45]. We then compute the L most important 3×3 patches, where L is the number of patches queried by IP for that particular input image. For computing the attribution/importance of a patch we average the attributions of all the pixels in that patch (following [20]). We proceed as follows: *(i)* Given $e(x)$, compute the patch with maximum attribution and add these pixels to our explanation, *(ii)* Zero-out the attributions of all the pixels in the previously selected patch and repeat step *(i)* until L patches are selected. The final explanation consists of L possibly overlapping patches. Now, we evaluate the sufficiency of the generated explanation for the model's prediction by estimating the MAP accuracy of the posterior over labels given the intensities in the patches included in this explanation. This is done via a VAE trained to learn the joint distribution over image pixels and class labels. We experiment with both the raw attribution scores returned by IG and DeepSHAP and also the absolute values of the attribution scores for $e(x)$. The results are reported in Table 2. In almost all cases (with the exception of DeepSHAP on Fashion-MNIST), IP generates explanations that are more predictive of the class label than popular attribution methods.

6 Conclusion

We have presented a step towards building trustworthy interpretable machine learning models that respect the domain- and user-dependent nature of interpretability. We address this by composing user-defined, interpretable queries into concise explanations. Furthermore, unlike many contemporary attempts at explainability, our method is not post-hoc, but is *interpretable by design* and guaranteed to produce faithful explanations. We formulate a tractable approach to implement this framework through deep generative models, MCMC algorithms, and the information pursuit algorithm. Finally, we demonstrate the effectiveness of our method across various vision and language tasks at generating concise explanations describing the underlying reasoning process behind the prediction. Future work will be aimed at extending the proposed framework to more complex tasks beyond classification such as scene parsing, image captioning, and sentiment analysis.

Acknowledgments

The authors thank María Pérez Ortiz and John Shawe-Taylor for their contributions to the design of the experiments on document classification presented in Section 5.0.3.

References

[1] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "Xai-explainable artificial intelligence," *Sci. Robot.*, vol. 4, no. 37, 2019, Art. no. eaay7120.
[2] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019.
[3] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Interpretable machine learning: Definitions, methods, and applications," 2019, *arXiv:1901.04592*.
[4] European Commission, "Building trust in human-centric artificial intelligence," Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions: COM (2019) 168 final 8.4, 2019.
[5] United States Food and Drug Administration, Virtual public workshop - transparency of artificial intelligence/machine learning-enabled medical devices," Oct. 14, 2021. [Online]. Available: https://www.fda.gov/media/154423/download
[6] U. Johansson, C. Sönströd, U. Norinder, and H. Boström, "Trade-off between accuracy and interpretability for predictive in silico modeling," *Future Med. Chem.*, vol. 3, no. 6, pp. 647–663, 2011.
[7] J. Wanner, L.-V. Herm, K. Heinrich, and C. Janiesch, "Stop ordering machine learning algorithms by their explainability! an empirical investigation of the tradeoff between performance and explainability," in *Proc. Conf. e-Bus., e-Serv. e-Soc.*, 2021, pp. 245–258.
[8] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. MicroElectronics*, 2018, pp. 0210–0215.
[9] A. B. Arrieta et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, 2020.
[10] D. Gunning and D. Aha, "Darpa's explainable artificial intelligence (XAI) program," *AI Mag.*, vol. 40, no. 2, pp. 44–58, 2019.
[11] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, 2010.
[12] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2013, *arXiv:1312.6034*.
[13] S. Kolek, D. A. Nguyen, R. Levie, J. Bruna, and G. Kutyniok, "A rate-distortion framework for explaining black-box model decisions," in *Proc. Int. Workshop Extending Explainable AI Beyond Deep Models Classifiers*, 2022, pp. 91–115.
[14] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3145–3153.
[15] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
[16] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.
[17] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: Removing noise by adding noise," 2017, *arXiv: 1706.03825*.

[18] A. Subramanya, V. Pillai, and H. Pirsiavash, "Fooling network interpretation in image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2020–2029.

[19] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9525–9536.

[20] M. Yang and B. Kim, "Benchmarking attribution methods with relative feature importance," 2019, *arXiv:1907.09701*.

[21] P.-J. Kindermans et al., "The (Un) reliability of saliency methods," in *Explainable AI: Interpreting, Explaining Visualizing Deep Learn.*, 2019, pp. 267–280.

[22] H. Shah, P. Jain, and P. Netrapalli, "Do input gradients highlight discriminative features?," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 2046–2059, 2021.

[23] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling lime and shap: Adversarial attacks on post hoc explanation methods," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, 2020, pp. 180–186.

[24] P. W. Koh et al., "Concept bottleneck models," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5338–5348.

[25] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, "This looks like that: Deep learning for interpretable image recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8930–8941.

[26] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, "Interpretable machine learning: Fundamental principles and 10 grand challenges," *Statist. Surv.s*, vol. 16, pp. 1–85, 2022.

[27] T. M. Janssen and B. H. Partee, "Compositionality," in *Handbook of Logic and Language*. Elsevier, 1997, pp. 417–473.

[28] H. Lakkaraju, S. H. Bach, and J. Leskovec, "Interpretable decision sets: A joint framework for description and prediction," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1675–1684.

[29] A. Wan et al., "{NBDT}: Neural-backed decision tree," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=mCLVeEpplNE

[30] J. Mu and J. Andreas, "Compositional explanations of neurons," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 17153–17163, 2020.

[31] E. Jahangiri, E. Yoruk, R. Vidal, L. Younes, and D. Geman, "Information pursuit: A Bayesian framework for sequential scene parsing," 2017, *arXiv:1701.02343*.

[32] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6541–6549.

[33] E. Hernandez, S. Schwettmann, D. Bau, T. Bagashvili, A. Torralba, and J. Andreas, "Natural language descriptions of deep visual features," 2022, *arXiv:2201.11114*.

[34] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Evanston, IL, USA: Routledge, 2017.

[35] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[36] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Comput.*, vol. 9, no. 7, pp. 1545–1588, 1997.

[37] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[38] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 161–168.

[39] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181, 2014.

[40] H. Xu et al., "When are deep networks really better than decision forests at small sample sizes, and how?," 2021, *arXiv:2108.13637*.

[41] P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulo, "Deep neural decision forests," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1467–1475.

[42] D. Geman and B. Jedynak, "An active testing model for tracking roads in satellite images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 1, pp. 1–14, 1996.

[43] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 839–847.

[44] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?" explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1135–1144.

[45] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 4768–4777.

[46] B. Kim et al., "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV)," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2668–2677.

[47] B. Zhou, Y. Sun, D. Bau, and A. Torralba, "Interpretable basis decomposition for visual explanation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 119–134.

[48] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, and P. Ravikumar, "On completeness-aware concept-based explanations in deep neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 20554–20565, 2020.

[49] D. Alvarez Melis and T. Jaakkola, "Towards robust interpretability with self-explaining neural networks," *Adv. Neural Inf. Process. Syst.*, 2018, pp. 7786–7795.

[50] M. Bohle, M. Fritz, and B. Schiele, "Convolutional dynamic alignment networks for interpretable classifications," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10 029–10 038.

[51] M. Wu, S. Parbhoo, M. C. Hughes, V. Roth, and F. Doshi-Velez, "Optimizing for interpretability in deep neural networks with tree regularization," *J. Artif. Intell. Res.*, vol. 72, pp. 1–37, 2021.

[52] V. Pillai and H. Pirsiavash, "Explainable models with consistent interpretations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 3, 2021, pp. 2431–2439.

[53] Z. Chen, Y. Bei, and C. Rudin, "Concept whitening for interpretable image recognition," *Nat. Mach. Intell.*, vol. 2, no. 12, pp. 772–782, 2020.

[54] M. De-Arteaga et al., "Bias in bios: A case study of semantic representation bias in a high-stakes setting," in *Proc. Conf. Fairness, Accountability, Transparency*, 2019, pp. 120–128.

[55] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4291–4308, Oct. 2021.

[56] D. Pruthi, M. Gupta, B. Dhingra, G. Neubig, and Z. C. Lipton, "Learning to deceive with attention-based explanations," 2019, *arXiv:1909.07913*.

[57] M. Craven and J. Shavlik, "Extracting tree-structured representations of trained networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 1995, pp. 24–30.

[58] D. Dancey, D. A. McLean, and Z. A. Bandar, "Decision tree extraction from trained neural networks," in *Proc. 19th Conf. Artif. Intell.*, 2004, pp. 515–519.

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

[60] A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5506–5514.

[61] U. C. Biçici, C. Keskin, and L. Akarun, "Conditional information gain networks," in *Proc. 24th Int. Conf. Pattern Recognit.*, 2018, pp. 1390–1395.

[62] V. N. Murthy, V. Singh, T. Chen, R. Manmatha, and D. Comaniciu, "Deep decision network for multi-class image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2240–2248.

[63] R. Vilalta, G. Blix, and L. Rendell, "Global data analysis and the fragmentation problem in decision tree induction," in *Proc. Eur. Conf. Mach. Learn.*, 1997, pp. 312–326.

[64] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," 2000, *arXiv:physics/0004057*.

[65] K. Chaloner and I. Verdinelli, "Bayesian experimental design: A review," *Statist. Sci.*, vol. 10, no. 3, pp. 273–304, 1995.

[66] R. Sznitman and B. Jedynak, "Active testing for face detection and localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1914–1920, Oct. 2010.

[67] M. Cuturi, O. Teboul, Q. Berthet, A. Doucet, and J.-P. Vert, "Noisy adaptive group testing using Bayesian sequential experimental design," 2020, *arXiv:2004.12508*.

[68] S. Branson, G. Van Horn, C. Wah, P. Perona, and S. Belongie, "The ignorant led by the blind: A hybrid human–machine vision system for fine-grained categorization," *Int. J. Comput. Vis.*, vol. 108, no. 1, pp. 3–29, 2014.

[69] V. Mnih et al., "Recurrent models of visual attention," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2204–2212.

[70] G. Elsayed, S. Kornblith, and Q. V. Le, "Saccader: Improving accuracy of hard attention models for vision," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 702–714.

[71] M. Li, S. S. Ge, and T. H. Lee, "Glance and glimpse network: A stochastic attention model driven by class saliency," in *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 572–587.
[72] H. Li, P. Wang, C. Shen, and A. v. d. Hengel, "Visual question answering as reading comprehension," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6319–6328.
[73] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A deep learning approach to visual question answering," *Int. J. Comput. Vis.*, vol. 125, no. 1, pp. 110–135, 2017.
[74] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," 2019, *arXiv:1904.12584*.
[75] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4613–4621.
[76] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 289–297.
[77] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 39–48.
[78] H. Laurent and R. L. Rivest, "Constructing optimal binary decision trees is np-complete," *Inf. Process. Lett.*, vol. 5, no. 1, pp. 15–17, 1976.
[79] M. I. Belghazi et al., "MINE: Mutual information neural estimation," 2018, *arXiv:1801.04062*.
[80] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, *arXiv:1312.6114*.
[81] A. Doucet et al., "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook Nonlinear Filtering*, vol. 12, no. 656–704, 2009, Art. no. 3.
[82] A. Jalal, S. Karmalkar, A. Dimakis, and E. Price, "Instance-optimal compressed sensing via posterior sampling," in *Proc. Mach. Learn. Res.*, 2021, pp. 4709–4720.
[83] E. Nijkamp, M. Hill, T. Han, S.-C. Zhu, and Y. N. Wu, "On the anatomy of MCMC-based maximum likelihood learning of energy-based models," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 5272–5280.
[84] A. Durmus and E. Moulines, "High-dimensional Bayesian inference via the unadjusted langevin algorithm," *Bernoulli*, vol. 25, no. 4A, pp. 2854–2882, 2019.
[85] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 681–688.
[86] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[87] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[88] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha, "Deep learning for classical japanese literature," 2018, *arXiv:1812.01718*.
[89] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200–2011 dataset," 2011.
[90] R. Misra, "News category dataset," 2022, *arXiv:2209.11429*.
[91] M. Lavin, "Analyzing documents with TF-IDF," *Program. Historian*, 2019.
[92] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[93] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
[94] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3319–3328.

Aditya Chattopadhyay received the bachelor's of technology degree in computer science and the master's of science by research degree in computational natural sciences from the International Institute of Information Technology, Hyderabad, in 2016 and 2018, respectively. He is currrently working toward the PhD degree with the Computer Science Department, Johns Hopkins University. His research interests include explainable AI, probabilistic graphical models and Bayesian inference.

Stewart Slocum received the BS degree in computer science and applied mathematics from Johns Hopkins University, in 2021. His research interests center on principled deep learning methods with performance and robustness guarantees.

Benjamin D. Haeffele received the BS degree in electrical engineering from the Georgia Institute of Technology, in 2006 and the PhD degree in biomedical engineering from Johns Hopkins University, in 2015. He is an associate research scientist with the Mathematical Institute for Data Science at Johns Hopkins University. His research interests involve developing theory and algorithms for processing high-dimensional data at the intersection of machine learning, optimization, and computer vision. In addition to basic research in data science he also works on a variety of applications in medicine, microscopy, and computational imaging.

René Vidal (Fellow, IEEE) received the BS degree in electrical engineering (valedictorian) from the Pontificia Universidad Católica de Chile, in 1997, and the MS and PhD degrees in electrical engineering and computer science from the University of California at Berkeley, in 2000 and 2003, respectively. He is currently the director of the Mathematical Institute for Data Science (MINDS) and the Hershel L. Seder professor of Department of Biomedical Engineering, The Johns Hopkins University, where he has been since 2004. He is co-author of the book "Generalized Principal Component Analysis" (Springer 2016), co-editor of the book "Dynamical Vision" (Springer 2006) and co-author of more than 300 articles in machine learning, computer vision, signal and image processing, biomedical image analysis, hybrid systems, robotics and control. He is or has been associate editor in chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *Computer Vision and Image Understanding*, associate editor or guest editor of *Medical Image Analysis*, the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *SIAM Journal on Imaging Sciences*, *Computer Vision and Image Understanding*, the *Journal of Mathematical Imaging and Vision*, the *International Journal on Computer Vision and Signal Processing Magazine*. He has received numerous awards for his work, including the 2021 Edward J. McCluskey Technical Achievement Award, the 2016 D'Alembert Faculty Fellowship, the 2012 IAPR J.K. Aggarwal Prize, the 2009 ONR Young Investigator Award, the 2009 Sloan Research Fellowship and the 2005 NSF CAREER Award. He is a fellow of IAPR, fellow of AIMBE, and a member of the ACM and SIAM.

Donald Geman (Life Senior Member, IEEE) received the BA degree in literature from the University of Illinois and the PhD degree in mathematics from Northwestern University. He was a distinguished professor with the University of Massachusetts until 2001, when he joined the Department of Applied Mathematics and Statistics, Johns Hopkins University, where he is currently a member of the Center for Imaging Science and the Institute for Computational Medicine. His current research interests include statistical learning, computer vision, and computational biology. He is a member of the National Academy of Sciences and a fellow of the IMS and SIAM.