Near-Optimal Machine Teaching via Explanatory Teaching Sets

Yuxin Chen
Caltech

Oisin Mac Aodha
Caltech

Shihan Su
Caltech

Pietro Perona
Caltech

Yisong Yue
Caltech

Abstract

Modern applications of machine teaching for humans often involve domain-specific, non-trivial target hypothesis classes. To facilitate understanding of the target hypothesis, it is crucial for the teaching algorithm to use examples which are interpretable to the human learner. In this paper, we propose NOTES, a principled framework for constructing interpretable teaching sets, utilizing *explanations* to accelerate the teaching process. Our algorithm is built upon a natural stochastic model of learners and a novel submodular surrogate objective function which greedily selects interpretable teaching examples. We prove that NOTES is competitive with the optimal explanation-based teaching strategy. We further instantiate NOTES with a specific hypothesis class, which can be viewed as an interpretable approximation of *any* hypothesis class, allowing us to handle complex hypothesis in practice. We demonstrate the effectiveness of NOTES on several image classification tasks, for both simulated and real human learners. Our experimental results suggest that by leveraging explanations, one can significantly speed up teaching.

1 Introduction

How can we design the best strategy to teach a complex hypothesis to a novice? How can we explain a concept via a representative, yet interpretable, set of labeled examples such that the concept derived from this set also generalizes well to unseen examples? Such problems, also known as machine teaching, have been studied in various application domains, including model compression (Ba & Caruana, 2014; Romero et al., 2014), citizen science and crowdsourcing services (Sullivan et al., 2009), and human-in-the-loop systems (Cakmak & Thomaz, 2014; Johns et al., 2015).

Unlike most machine learning applications where the *informativeness* of training examples plays a vital role, machine teaching requires the examples selected by the teaching algorithm to be *interpretable*, because the information contained in an example is useful *only if* it is easily accessible to the learner. There could be many factors that prevent a learner from adopting a novel concept; two notable examples are (1) hard teaching examples and (2) complex hypothesis classes. As an example for the former, in image classification, a learner could be overwhelmed by the information contained in a training image and ignore the "important pieces" of information which may be crucial for determining the label. In the case of learning over complex hypothesis classes, a learner may have converged to a suboptimal hypothesis, which makes it hard for her to switch to a different one unless she receives an interpretable explanation. Therefore, as the complexity of the teaching tasks increases, it becomes even more challenging to convey the teacher's hypothesis.

While the label-based teaching strategies quickly become incapable of handling such complex tasks, it has been shown that additional information (such as highlighting regions or features on an image) can help improve novice classification performance by guiding the student's attention (Grant & Spivey, 2003; Roads et al., 2016; Mac Aodha et al., 2018). Having an additional degree of expressiveness in the feedback allows us to generate more intuitive explanations of the teacher's predictive model, and hence dramatically improves the learner's ability to learn a new concept. While these results are encouraging, several important questions remain open. First, allowing explanations increases the computational complexity of finding the optimal teaching set, by orders of magnitude, as there are many more available actions for the teacher to choose from. Second, this flexibility greatly improves our ability to properly model the learners. The explanations must be constructed in a way that is not only informative about the teacher's hypothesis, but also makes sense to the learner. Third, explanations often involve

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. JMLR: W&CP volume 7X. Copyright 2018 by the author(s).

structured information, i.e., each training example may have multiple interacting features which are informative for its label. Thus far, there has been little theoretical understanding on how to efficiently construct the set of teaching examples for complex, structured hypotheses.

Our contribution. We explore both theoretical and practical aspects of the explanation-based teaching problem. First, we propose a novel formalism of the teaching problem, where we model the learners' hypothesis as a two-stage decision-making process. Given a teaching example, e.g., an image, she first chooses which part of the image to inspect, and then makes a prediction of the label based on that part information. By separating out the learner's attention model from her decision model, we can thus treat the explanations provided by the teacher as intermediate labels of the teaching example, and lift existing algorithmic tools for label-based machine teaching to handle the explanatory feedback. Based on such formalism, we develop a surrogate objective function and prove that the greedy teaching strategy, which we call NOTES (*Near-Optimal Teaching via Explanatory Sets*), achieves strong theoretical guarantees.

We then instantiate NOTES with a particular hypothesis class for the learners, which can be viewed as an approximation of *any* target hypothesis class. The hypothesis class we adopt in this paper is based on recent advances in interpretable models (Ribeiro et al., 2016). In particular, each hypothesis is an ensemble of locally interpretable models, each of which is trained to approximate a global classifier in a local neighborhood. Our construction of hypothesis class enables us to deal with complex hypotheses and data distributions in a practical manner (i.e., beyond linear classifiers). Moreover, since the teacher's ground truth classification model can also be approximated by a hypothesis in our hypothesis class, we can automatically generate (new) explanations of the labels provided by the teacher, even if they are not available in the teaching set.

We empirically evaluate our teaching strategy on simulated learners on five different datasets, two of which involving synthetic visual objects and three on real-world images. We further conduct experiments with real-world user studies on Amazon Mechanical Turk. Our results show the clear advantage of using NOTES as a principled approach for handling explanatory feedback.

2 Related Work

Machine teaching. Machine teaching has shown thriving development in the past two decades, in both the practical and theoretical machine teaching communities. In practice, intelligent tutoring systems have been developed to teach diverse topics such as mathematics (Koedinger et al., 1997; Canfield, 2001) and languages (von Ahn, 2013). A rich class of teaching approaches has been explored to model the potential of the learners, including, among others, heuristic-based approaches (Basu & Christensen, 2013), Bayesian models (Corbett & Anderson, 1994; Eaves et al., 2015), recurrent neural networks (Piech et al., 2015), and reinforcement learning based models (Rafferty et al., 2011; Bak et al., 2016; Whitehill & Movellan, 2017).

From the theoretical perspective, there have been many attempts to make the sequence of teaching labels optimal. The notion of teaching "optimality" has been rigorously defined in terms of the teaching dimension (Goldman & Kearns, 1995), or via cognitive models (Patil et al., 2014) that consider human learner's learning capacity as a constraint. However, existing algorithms are commonly analyzed in terms of their statistical, as opposed to computational, complexity (Goldman & Kearns, 1995; Zhu, 2013; Chen et al., 2016; Liu et al., 2017). An alternative line of work considers machine teaching as a discrete optimization problem. Here, the goal is to efficiently identify a minimal number of informative examples to teach some concept. While the optimal solution is intractable in general, it is of great interest to devise efficient algorithms that are competitive with the optimal algorithm. Along with this line, Singla et al. (2014) proposed a greedy framework for teaching visual classification tasks, where they rely on a submodular surrogate function that enables efficient optimization. While our work is related in spirit, it provides richer information and interpretable feedback to the learners.

Interpretable models. Interpretable predictive modeling, in general, is useful in many application domains where black-box models are hard to interpret (e.g., medical/biological research, emergency response planning, business processes, etc.). Depending on the heuristics used to measure interprebility, existing methods can be categorized into three types: (1) sparse models such as sparse linear classifiers, which use a small number of features and parameters (Chang et al., 2009; Ustun & Rudin, 2014; Ribeiro et al., 2016), (low-rank) latent factor models, (2) discretization based methods such as decision trees and decision lists, which split up a problem into several (independent) sub-problems (Letham et al., 2015; Lakkaraju et al., 2016), and (3) prototype-based classifiers such as nearest neighbors, which use examples and basic features instead of formal models and constructed features (Cover & Hart, 1967). In our work, we view machine teaching as the problem of developing an interpretable model. Rather than building an *offline* interpretable model to represent the hypotheses, we would like to generate a sequence of interpretable predictions (i.e., explanations) at *run-time*, so that the learner would fully adapt to (i.e., interpret) the teacher's hypothesis. Our approach builds upon existing interpretable models as a means of generating interpretable explanations.

3 Problem Statement

Here we formally state the problem studied in this paper and introduce some notation for our model.

3.1 Teaching set and hypothesis class

Suppose we are given a ground (teaching) set of examples $\mathcal{X} := \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$. Let $\mathcal{Y} = \{y_1, \ldots, y_c\}$ be the set of all possible class labels. Each example $\mathbf{x}_i \in \mathcal{X}$ has a label $y_i \in \mathcal{Y}$, and (up to) k interpretable *attributes* $\mathbf{a}_i := (a_{i1}, \ldots, a_{ik})$. Let $\mathbb{A} := \mathbb{R} \cup \{\mathsf{null}\}$ denote the domain of the attribute values[1], where $a_{ij} \in \mathbb{R}$ if the j^{th} attribute is present, and $a_{ij} = \mathsf{null}$ otherwise. Furthermore, let us use $\phi : \mathcal{X} \to \mathbb{A}^k$ to denote the process by which attributes are generated. We can think of these attributes as features that can be easily parsed and interpreted by the learners (in other words, ϕ is known to the learners, and the output of ϕ can be informed by the learners' knowledge of the world). For example, in medical diagnosis, the attributes may represent patients' medical records, such as gender, age, and outcomes of medical tests; in image classification, the attributes could correspond to high-level visual features such as image parts or natural language descriptions of the image.

Attention model. The number of attributes k can be large. However, we assume that for each $\mathbf{x}_i$, a *sparse* subset of these attributes is sufficient to determine its label y_i. Let $f : \mathcal{X} \to \mathbb{B}^k$ be a (deterministic) mapping from $\mathbf{x}_i \in \mathcal{X}$ to binary vectors of length k, denoted by $\mathbf{e}_i := \{e_{i1}, \ldots, e_{ik}\}$, indicating which attributes are important in determining the label for $\mathbf{x}_i$. Intuitively, f can be viewed as an attention model of the learner that captures which part of the attribute space one should focus on to learn the classification task.

Decision model. Once the attributes are chosen, we can decide upon the label of an example. We define the decision model $g : (\mathbb{A} \times \mathbb{B})^k \to \mathcal{Y}$ as a deterministic mapping from "important" attributes to class labels. Therefore, our end-to-end decision model of the learner, denoted by a function $h : \mathcal{X} \to \mathcal{Y}$, can be expressed as a composite of the attention model and the decision model:

$$h(\mathbf{x}) := g(\phi(\mathbf{x}), f(\mathbf{x})). \tag{3.1}$$

We refer to each possible model of the learner h as a *hypothesis*, and the space of all possible decision models as the hypothesis class, denoted by $\mathcal{H}$.

3.2 Teaching protocol and model of learner

We focus on the *non-adaptive* setting, where the teacher (sequentially) shows a subset of the teaching examples to the learner, without observing the learner's prediction of the examples shown to them. For each teaching example revealed, the teacher provides both its *label* and an *explanation* of why the label is assigned. We assume that the teacher has access to such information for all of the examples in $\mathcal{X}$, as well as a prior distribution over the learner's hypotheses $\mathbb{P}[h]$.

Learner model with label-only instructions. To characterize how learners adapt to the examples received from the teacher, we adopt the stochastic learner model introduced by Singla et al. (2014). Their model was originally proposed in the context of teaching with label-only feedback to the learner. In their setting, at the beginning of a teaching session, the learner randomly chooses a hypothesis $h \in \mathcal{H}$ according to $\mathbb{P}[h]$. When receiving a new example, the learner either agrees or disagrees with its true label (as provided by the teacher). If the true label is consistent with the learner's current hypothesis, she sticks to the same hypothesis and proceeds to the next round. Alternatively, if it is inconsistent, she randomly switches to a hypothesis in $\mathcal{H}$ according to a distribution which reduces the probability of the hypotheses that are inconsistent with the true labels.

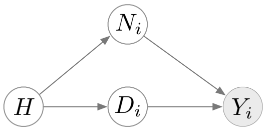


Figure 1: The label-based learner model of Singla et al. (2014). Node Y_i is observable (by the learner) while others are unobserved.

The above stochastic process can be equivalently represented by the graphical model shown in Fig. 1. Let $H \in \mathcal{H}$, $Y_i \in \mathcal{Y}$ be random variables representing the learner's hypothesis and her prediction of the label of example $\mathbf{x}_i$, respectively. We assume that Y_i depends on H and another boolean random variable $N_i \in \mathbb{B}$, which encodes the presence of noise. First, $\mathbf{x}_i$ goes through the deterministic mapping H as defined in Eq. (3.1). The output, denoted by D_i, will then be perturbed by the noise if $N_i = \mathsf{True}$, and produce Y_i; otherwise (if $N_i = \mathsf{False}$) $Y_i = D_i$. Importantly, the noise N_i depends on h and is *asymmetric*:

$$\mathbb{P}[N_i = \mathsf{True} \mid H = h] = \begin{cases} 0 & \text{if } h(\mathbf{x}_i) = y_i \\ \nu_i & \text{o.w.,} \end{cases}$$

where $\nu_i \in [0, 1]$ is some noise parameter known to the learner. Intuitively, given example $\mathbf{x}_i$, the learner explains her consistent prediction as "my hypothesis is correct", and inconsistent prediction as "it is probably caused by noise". Applying Bayes' rule (and marginalizing over N_i), we get

$$\mathbb{P}[Y_i = y_i \mid H = h] = \begin{cases} 1 & \text{if } h(\mathbf{x}_i) = y_i \\ \nu_i & \text{o.w.} \end{cases} \tag{3.2}$$

[1] For interpretable models such as decision trees, a_{ij}'s are boolean variables (i.e., $\mathbb{A} \equiv \mathbb{B}$), indicating whether a decision rule (i.e., predicate) is satisfied.

Let $\mathcal{S} \subseteq \{1,\dots,m\}$ be the index set of the examples received by the learner. We use $\mathbf{x}_{\mathcal{S}} := \cup_{i\in\mathcal{S}}\{\mathbf{x}_i\}$, $y_{\mathcal{S}} := \cup_{i\in\mathcal{S}}\{y_i\}$ to denote the selected examples and their (observed) true labels respectively. Upon observing $y_{\mathcal{S}}$, the learner draws her hypothesis from the posterior distribution of H, defined by

$$\mathbb{P}[H=h\,|\,y_{\mathcal{S}}] \propto \mathbb{P}[h]\prod_{i\in\mathcal{S}}\mathbb{P}[y_i\,|\,h] = \mathbb{P}[h]\prod_{i\in\mathcal{S}\wedge h(\mathbf{x}_i)\neq y_i}\nu_i.$$

As a result, hypotheses that are "more inconsistent" with the received labels are less likely to be chosen, which reflects how the learner learns from the examples.

Explanation-based learner model. In §3.1, we formulated the learner's attention as a function of the teaching example. The output of the attention function, by construction, is interpretable and can naturally be used as an "explanation" for the class label.

Our explanation-based learner model is depicted in Fig. 2. Let F be a random variable representing the learner's attention function f. For a given input $\mathbf{x}_i$, we use random variable[2] $E_i \in \mathbb{B}^k$ to denote the explanation (as a $k-$dimensional boolean vector) of its class label Y_i. The value of E_i is determined by the true attention function which is unknown to the learner, as well as another random variable N_i^f indicating the noise in the estimated attention vector for $\mathbf{x}_i$. Similarly, let G be a random variable representing the learner's decision function g, and N_i^g be the noise involved in the decision. The learner's prediction of Y_i, therefore, depends on the explanation E_i, the (output of) the decision function G and the noise N_i^g.

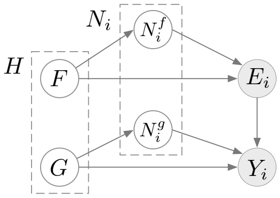


Figure 2: Our explanation-based model of the learners. The learner's hypothesis H is now decomposed into two independent random variables, namely the attention function F and the decision function G. Each random variable is perturbed by its associated noise before generating the label of the teaching example $\mathbf{x}_i$.

Intuitively, the explanation E_i can be viewed as an intermediate label of the input example. Following Eq. (3.2) for the label-only scenario, we parameterize our explanation-based learner model as

$$\mathbb{P}[E_i=\mathbf{e}_i\,|\,F=f] = \begin{cases} 1 & \text{if } f(\mathbf{x}_i)=\mathbf{e}_i \\ \nu_i^f & \text{o.w.} \end{cases} \tag{3.3}$$

$$\mathbb{P}[Y_i=y_i\,|\,E_i=\mathbf{e}_i, G=g] = \begin{cases} 1 & \text{if } g(\phi(\mathbf{x}_i),\mathbf{e}_i)=y_i \\ \nu_i^g & \text{o.w.,} \end{cases}$$

where ν_i^g (resp. ν_i^f) is some known parameter indicating the noise level of false predictions of the label (resp. explanation).

3.3 Explanation-based machine teaching

Similar to the label-based teaching scenario, we would like to design a strategy for selecting useful teaching examples (i.e., with both their *labels* and *explanations*), such that the resulting distribution of the learner's hypotheses implies a low prediction error on the labels for all future examples. More concretely, let $\text{err}(h) := \frac{|\{\mathbf{x}_i\in\mathcal{X}:h(\mathbf{x}_i)\neq y_i\}|}{m}$ be the error rate of hypothesis h. Assume that we have access to the prior distributions of the learner's attention and decision models $\mathbb{P}[f]$ and $\mathbb{P}[g]$, as well as the conditional probabilities (i.e., the noise parameters) ν_i^f, ν_i^g for all f, g in the support of the prior distribution with $i\in\{1,\dots,m\}$. Our goal is to find a set $\mathcal{S}^*$ of the minimal size, such that upon observing the labels and explanations of $\mathbf{x}_{\mathcal{S}^*}$, the learner would achieve an expected error rate of at most ϵ. Formally, we seek

$$\mathcal{S}^* \in \underset{\mathcal{S}\subseteq\{1,\dots,m\}}{\text{argmin}}\ |\mathcal{S}|, \text{ s.t., } \mathbb{E}[\text{err}(h)\,|\,\mathbf{e}_{\mathcal{S}}, y_{\mathcal{S}}] \leq \epsilon. \tag{3.4}$$

where $\mathbf{e}_{\mathcal{S}} := \cup_{i\in\mathcal{S}}\{\mathbf{e}_i\}$ and $y_{\mathcal{S}}$ denote the explanations and labels for the set of examples indexed by $\mathcal{S}$.

It can be shown that Problem 3.4 is NP-hard even for the special case of label-based teaching scenario (Singla et al., 2014). Even though adding explanations further increases the *computation complexity* of selecting the optimal teaching set, we show that it can drastically reduce the teaching effort (i.e., *label complexity*).

Proposition 1. *There exist problem instances where the optimal label-based strategy is arbitrarily worse than the optimal explanation-based strategy.*

4 The NOTES Algorithm

Let us use f^*, g^* and $h^* = g^*\circ f^*$ to denote the *true* attention function, decision function, and hypothesis respectively. That is, f^* and g^* output the true explanation and label for any teaching example (i.e., $\forall i$, $f^*(\mathbf{x}_i)=\mathbf{e}_i$ and $g^*(\phi(\mathbf{x}_i),\mathbf{e}_i)=y_i$). Note that teaching the true hypothesis h^* amounts to teaching the learner *not to* choose hypotheses that make an error. A sufficient condition for an arbitrary $h=g\circ f$ to make an error on example $\mathbf{x}_i$

[2] From the learner's perspective, both E_i and Y_i are unknown until the teacher reveals them, and therefore are considered as random variables (even though the true explanations and labels are fixed from the teacher's perspective).

is that either f or g makes an error at $\mathbf{x}_i$, but not both[3]:

$$\begin{aligned} &g\circ f(\mathbf{x}_i)\neq y_i \text{ if}\\ &\quad (f(\mathbf{x}_i)\neq f^*(\mathbf{x}_i)\wedge g(\phi(\mathbf{x}_i),\mathbf{e}_i)=g^*(\phi(\mathbf{x}_i),\mathbf{e}_i)) \vee\\ &\quad (f(\mathbf{x}_i)=f^*(\mathbf{x}_i)\wedge g(\phi(\mathbf{x}_i),\mathbf{e}_i)\neq g^*(\phi(\mathbf{x}_i),\mathbf{e}_i)). \end{aligned}$$

Meanwhile, a necessary condition for h to make an error on $\mathbf{x}_i$ is that at least one of its components makes an error:

$$\begin{aligned} &g\circ f(\mathbf{x}_i)\neq y_i \textit{ only} \text{ if}\\ &\qquad f(\mathbf{x}_i)\neq f^*(\mathbf{x}_i)\vee g(\phi(\mathbf{x}_i),\mathbf{e}_i)\neq g^*(\phi(\mathbf{x}_i),\mathbf{e}_i). \end{aligned}$$

If we can effectively discourage the learner from choosing the wrong combinations of f and g's, we will make good progress towards teaching the learner h^*. Such intuition motivates us to consider a bipartite graph representation of the hypotheses (see Fig. 3), where we draw an edge between an attention function node and a decision function node as long as (exactly) one of them is true. In other words, each edge is associated with a node, either representing an attention function or a decision function, that has to be ruled out if we want to teach the learner h^*.

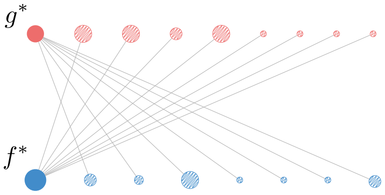


Figure 3: Illustration of the bipartite graph constructed by NOTES (Algorithm 1). In this figure, we see 9 decision functions (red nodes) and 8 attention functions (blue nodes). The size of each node represents the prior probability. Edges are drawn between the true decision functions (solid red) and all attention functions and between the true attention functions (solid blue) and all decision functions.

Formally, denote the set of edges as $\mathcal{E}:=\{\{f,g\}:(f=f^*\wedge g\neq g^*)\vee(f\neq f^*\wedge g=g^*)\}$. We define the initial weight of each edge $\{f,g\}\in\mathcal{E}$ as

$$w(\{f,g\}):=\mathbb{P}[f]\mathbb{P}[g](\mathrm{err}(g^*\circ f)+\mathrm{err}(g\circ f^*)). \tag{4.1}$$

Intuitively, one can think of $w(\{f,g\})$ as a measure of how much f and g are accountable for the expected error of the learner (with respect to the prior distribution). After the teacher picks examples $\mathcal{S}$ and shows $\{(\mathbf{x}_i,\,y_i,\,\mathbf{e}_i)\}_{i\in\mathcal{S}}$ to the learner, we *discount* the weight of the edge $\{f,\,g\}$ through Bayesian updates, i.e., by multiplying the probabilities of its incident nodes with the likelihood of the observations:

Algorithm 1: NOTES

```
Input: Teaching set $\{(\mathbf{x}_i,y_i,\mathbf{e}_i)\}_{i=1:m}$; prior $\mathbb{P}[f]$, $\mathbb{P}[g]$;
  noise parameters $\{\nu_i^f,\nu_i^g\}_{i=1:m}$; tolerance $\epsilon$.
begin
    $\mathcal{S}\leftarrow\emptyset$;
    while $r(\mathcal{S})>\mathbb{P}[f^*]\mathbb{P}[g^*]\epsilon$ do
        $i^*\leftarrow\mathrm{argmin}_i r(\mathcal{S}\cup\{i\})$
        $\mathcal{S}\leftarrow\mathcal{S}\cup\{i^*\}$
    Output: Selected teaching examples $\mathcal{S}$.
```

$w(\{f,g\}\,|\,\mathcal{S}):=w(\{f,g\})\cdot\mathbb{P}[\mathbf{e}_{\mathcal{S}},y_{\mathcal{S}}\,|\,f,g]$. Our objective function, denoted by r, is then defined as the remaining weight of the bipartite graph upon observing set $\mathcal{S}$:

$$\begin{aligned} r(\mathcal{S}) &:= \sum_{\{f,g\}\in\mathcal{E}} w(\{f,g\}\,|\,\mathcal{S})\\ &\overset{(a)}{=} \sum_{\{f,g\}\in\mathcal{E}} w(\{f,g\})\prod_{i\in\mathcal{S}}\mathbb{P}[\mathbf{e}_i,y_i\,|\,f,g]. \end{aligned}\tag{4.2}$$

Here, step (a) holds because $\mathbb{P}[\mathbf{e}_{\mathcal{S}},y_{\mathcal{S}}\,|\,f,g]$ can be factorized according to the graphical model in Fig. 2 (i.e., the tuples $(\mathbf{e}_i,y_i)$ and $(\mathbf{e}_j,y_j)$ are conditionally independent given g and f). A key property of the function r (more precisely, the reduction in r: $r(\emptyset)-r(\mathcal{S})$) that allows us to efficiently select teaching examples, as stated in Lemma 2, is that it satisfies submodularity.

Lemma 2. *The edge weight removed from the bipartite graph is a monotone submodular function of $\mathcal{S}$.*

Submodularity is a natural diminishing returns condition that makes r amenable to efficient greedy optimization. We now show that optimizing r provides a *sufficient* and *necessary* condition for optimizing the original objective function, i.e., the expected error probability. Concretely, we establish a connection between $r(\mathcal{S})$ and $\mathbb{E}[\mathrm{err}(h)\,|\,\mathbf{e}_{\mathcal{S}},y_{\mathcal{S}}]$ through the following lemma.

Lemma 3. *r is within constant factors of the expected error rate:*

$$\mathbb{P}[f^*]\mathbb{P}[g^*]\mathbb{E}[err(h)\,|\,\mathbf{e}_{\mathcal{S}},y_{\mathcal{S}}]\leq r(\mathcal{S})\leq\mathbb{E}[err(h)\,|\,\mathbf{e}_{\mathcal{S}},y_{\mathcal{S}}].$$

Lemma 3 implies that if $r(\mathcal{S})$ is sufficiently small, then the expected error of the learner is also small. Therefore, we can greedily select teaching examples according to $r(\mathcal{S})$ until $r(\mathcal{S})\leq\mathbb{P}[f^*]\mathbb{P}[g^*]\epsilon$, and by Lemma 3 we know that $\mathbb{E}[\mathrm{err}(h)\,|\,\mathbf{e}_{\mathcal{S}},y_{\mathcal{S}}]\leq\epsilon$. We call this greedy algorithm NOTES (Near-Optimal Teaching via Explanatory Sets) and outline it in Algorithm 1.

Theorem 4. *Let $OPT(\epsilon)$ be the worst-case cost of the optimal algorithm that achieves expected error rate of at most ϵ. The worst-case cost of NOTES achieving expected error rate ϵ is upper bounded by*

$$OPT(\frac{\mathbb{P}[f^*]\mathbb{P}[g^*]\epsilon}{2})\cdot\log\left(\frac{1}{\mathbb{P}[f^*]\mathbb{P}[g^*]\epsilon}\right).$$

[3] If both f and g make an error on $\mathbf{x}_i$, the composite function h may output a correct label for $\mathbf{x}_i$.

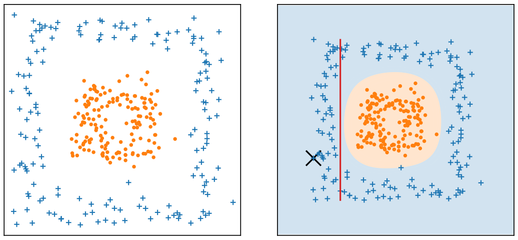
The proofs of Lemma 2, Lemma 3 and Theorem 4 are provided in the supplemental material. Observe that when the true attention function is known to the learner (i.e., $\mathbb{P}[f^*]=1$), Problem 3.4 reduces to the label-based teaching problem, in which case NOTES is equivalent to the STRICT policy proposed by Singla et al. (2014). Hence, our algorithm strictly generalizes STRICT to the explanation-based teaching scenario while still preserving provable guarantees.

5 Implementation Details

In §4, we introduced a general framework to accommodate explanations for machine teaching. We now instantiate NOTES with a specific class of attention and decision functions, and propose an efficient implementation for teaching such composite hypotheses.

We employ the LIME model of Ribeiro et al. (2016) to generate explanations for the examples in the teaching set. LIME was originally developed as an analytics tool for explaining the behavior of any machine learning classifier. It works by fitting simple, human-interpretable, models around local regions of the original data distribution. In particular, given an input example and its label produced by a complex hypothesis (say by the teacher's model h^*), LIME aims to reproduce the predictive results of h^* in the vicinity of the input via a sparse linear classifier (see Fig. 4). With only a small number of non-zero entries to inspect, the hope is that LIME can help humans understand h^*, at least, locally.

In the context of machine teaching, such interpretable approximations of the true model can naturally be used as explanations for the provided label. Given the original hypothesis class $\mathcal{H}$ and the teaching set $\mathcal{X}$, we can construct an approximate hypothesis class $\hat{\mathcal{H}}$, by running LIME on each $(h,\mathbf{x})$ pair. This naive construction strategy[4] gives us $m\cdot|\mathcal{H}|$ sparse linear functions (recall that $m=|\mathcal{X}|$ is the number of teaching examples).



(a) 2-D toy dataset (b) Local interpretation

Figure 4: Explaining a hypothesis with LIME on a 2-D toy dataset shown in Fig. 4(a). The teacher's hypothesis h^* is represented by the orange circle in Fig. 4(b). Given a teaching example (marked by "X"), the teacher first generates a local sparse approximation of h^* around it (as marked by the red line) as the explanation of its label. In words, the explanation can be interpreted as: "X is classified as the blue class, mainly because its horizon axis is less than the threshold marked by the red line".

Let us denote the sparse linear function generated from $(h,\mathbf{x})$ by $\ell_{(h,\mathbf{x})}$. Following the discussion from §3.1, we interpret $\ell_{(h,\mathbf{x})}$ as a two-stage decision-making function. First, it maps the input example $\mathbf{x}$ to the non-zero entries of its weight vector. Then, using the non-zero weights, it outputs a value that approximates h's prediction of the label of $\mathbf{x}$, i.e., $h(\mathbf{x})$ (as is defined in Eq. (3.1)). Since the non-zero dimensions of the weight vector naturally models the importance of the corresponding attributes, they can be used as explanations of the label. In other words, we can derive both the *output of the attention function* $f(\mathbf{x})$ and the *decision function* g itself from $\ell_{(h,\mathbf{x})}$.

Now that we have a concrete representation of f and g for each hypothesis of the learner, we are able to generate the full explanatory teaching set $\{(\mathbf{x}_i,y_i,\mathbf{e}_i)\}_{i=1:m}$, by applying the LIME approximation of the teacher's hypothesis to all teaching examples.[5] To run NOTES, we still need to know the learner's noise parameters ν_i^f,ν_i^g for each $\mathbf{x}_i$ (Eq. (3.3)). We propose to use the following likelihood functions for inconsistent explanations (labels) for $\mathbf{x}_i$:

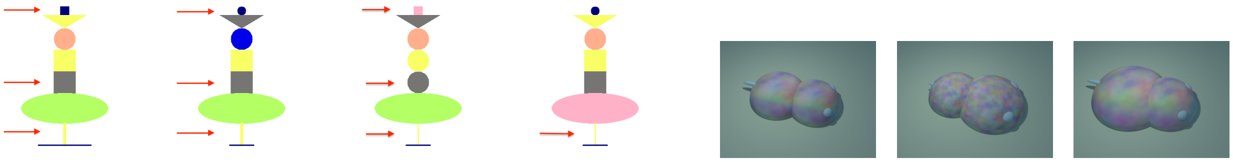
$$\begin{cases}\nu_i^f=2\mathrm{expit}\left(\alpha\left(\left|\frac{f_\ell(\mathbf{x}_i)\wedge\mathbf{e}_i}{f_\ell(\mathbf{x}_i)\vee\mathbf{e}_i}\right|-1\right)\right)\\ \nu_i^g=\mathrm{expit}\left(\beta\ell_{(h,\mathbf{x})}(\mathbf{x})y_i\right),\end{cases} \tag{5.1}$$

where f_ℓ represents the attention function associated with the linear function $\ell_{(h,\mathbf{x})}$ (i.e., the indices of the non-zero weights), $\mathrm{expit}(x)=\frac{1}{1+\exp(-x)}$ is the logistic function, and $\alpha,\beta>0$ are global parameters that can be tuned for different teaching tasks. Intuitively, α captures the learner's ability to adapt to the explanations, while β reflects the learner's ability to adapt to label feedback. If $\alpha,\beta\to\infty$, then $\nu_i^f=\nu_i^g=0$ which corresponds to the noise-free setting.

It is worth noting that, similar to h^*, a learner's hypothesis $h=g\circ f$ can also be an ensemble of sparse linear functions. In such cases, to compute the likelihood of a given teaching example $\mathbf{x}_i$, we use the sparse linear function $\ell_{(h,\mathbf{x})}$ whose associated example $\mathbf{x}$ being explained is the closest to the target teaching example $\mathbf{x}_i$, and apply Eq. (5.1) to compute ν_i^f and ν_i^g.

[4] One can also use an extension of LIME for explaining a global machine learning model (as opposed to a single prediction), reducing the number of linear functions used to approximate a hypothesis, see (Ribeiro et al., 2016).

[5] Admittedly, the performance of NOTES is limited by the quality of the underlying LIME local approximations. However, as we will see in §6, our algorithm is capable of making sensible explanations for most of the test scenarios.



(a) Mars 1 (b) Mars 2 (c) Jupiter 1 (d) Jupiter 2 (e) Weevil (f) Vespula 1 (g) Vespula 2

Figure 5: Sample images from the JM and VW datasets. For JM, objects from Mars must have blue color, grey square shape, and thick size on the three dimensions highlighted by the arrows, from top to bottom - otherwise they are from Jupiter. For VW, the Weevil has a mid-sized body and mid-sized head, while the Vespula does not.

6 Experiments

In this section, we present experimental results for both simulated learners and real human participants from Amazon Mechanical Turk.

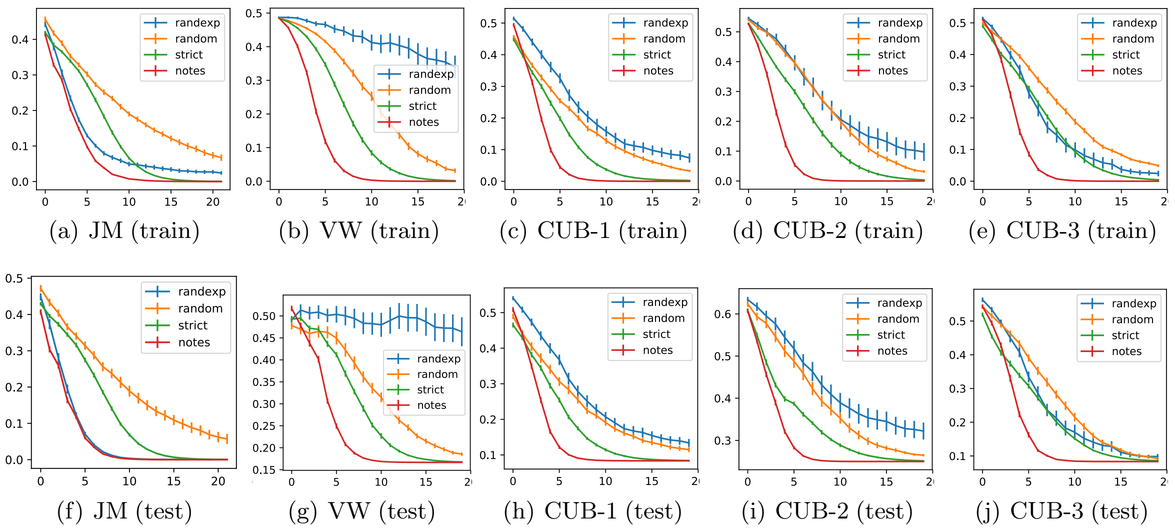
Baselines. We compare NOTES against three other baselines in our experiments: (1) random, which randomly selects a sequence of teaching examples, and only reveals their labels to the learners; (2) randexp, which randomly selects a teaching example, along with a random valid explanation to show to the learner; and (3) STRICT (Singla et al., 2014) which selects a teaching sequence with label-only feedback. We measure the overall performance of the different algorithms as the average classification error of the learners on the test set.

Datasets. We employed five different image datasets in our experiments - two synthetic and three real datasets. These datasets were selected with varying characteristics so that they reflect a wide range of practical scenarios. Our synthetic datasets, namely (1) "Jupiter vs Mars" (JM), and (2) "Vespula vs Weevil" (VW), each contains two groups of hypothetical alien objects with different features. We visualize these datasets by assembling meaningful visual parts defined by their feature values. Sample images of the two synthetic datasets are shown in Fig. 5. The JM dataset contains 128 images, each with 8 parts. Every part may have different color and shape, which gives us a 10-D binary feature vector (some parts share). The VW dataset was created based on the 2-D dataset shown in Fig. 4. The 2-D features are visualized as the head and body size of the hypothetical bugs (Vespula and Weevil), along with two non-informative dimensions - tail length and texture. This dataset is an extended version of a similar dataset proposed by (Singla et al., 2014). Unlike Singla et al. (2014), we use a non-linear decision rule to distinguish the two groups. This simulated data gives us full control over the generating distribution and the hypothesis space, and allows us to demonstrate the algorithms' teaching ability when we have full knowledge of the learners' hypotheses distribution. We further study three real bird species classification tasks in images. We select three pairs of visually similar species from the CUB dataset (Wah et al., 2011) - "*Heermann Gull vs Western Gull*" (CUB-1), "*Baird Sparrow* and *Chipping Sparrow*" (CUB-2), and "*Song Sparrow* vs *Northern Waterthrush*" (CUB-3). Each individual species consists of 60 images and we use all the visual attributes that have a corresponding part location in the images as our feature representation (237 out of 312). After grouping the attributes based on part locations, we are left with a 22-D categorical feature vector.

Hypothesis class. Unlike the simulated learners, for the CUB datasets, we do not have an estimate of the prior over the real learners' attention and decision functions. Therefore, we estimate the learner's hypothesis with a simple heuristic. Novices may make the wrong prediction of the bird species, but their mistakes are usually systematically consistent. Therefore for each dataset, we first divide it into six clusters and then randomly pick three clusters and assign them the same label - with the opposite label for the remaining three. This gives us $\binom{6}{3} = 20$ possible noisy label assignments for each dataset. We fit LIME to the perturbed datasets and generate 20 sparse linear classifiers for each of the perturbed label assignments. We further generate 20 hypotheses from the original dataset, where one of the hypotheses is the ensemble of all the sparse linear classifiers generated for each teaching example. This hypothesis is used as the ground-truth teacher's model for providing interpretable explanations to the learners, i.e., $h^* := g^* \circ f^*$. To guarantee stability during teaching, we remove all the teaching examples whose label are inconsistent with the prediction of h^*. Note that we only perform this preprocessing step for the teaching/training phase and use the original test set for evaluation. We set $\alpha = 1, \beta = 1$ for all experiments.

6.1 Experimental results

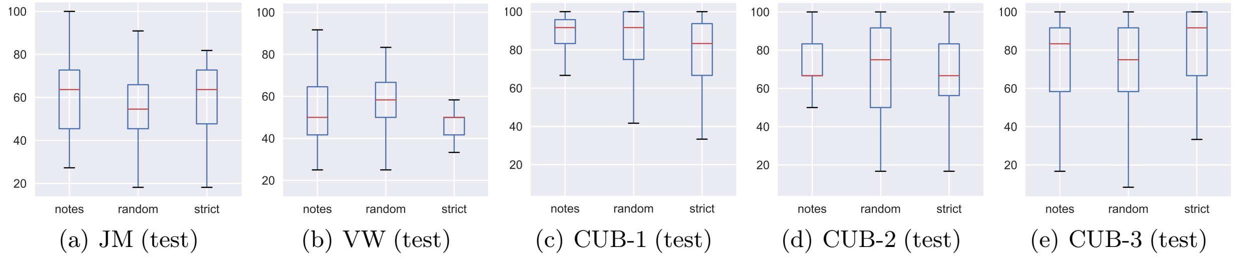
Simulated learners. We first conduct a study of NOTES with simulated learners on both the synthetic and real-world image datasets. For simulated learners, we know their hypothesis distribution, and thus these experiments allow us to closely inspect the behavior of the baseline algorithms. Experimental results, averaged over 30 repeats, are shown in Fig. 6. From these results, we can see that by incorporating explanations NOTES systematically outperforms all other baselines for simulated learners.



(a) JM (train) (b) VW (train) (c) CUB-1 (train) (d) CUB-2 (train) (e) CUB-3 (train)

(f) JM (test) (g) VW (test) (h) CUB-1 (test) (i) CUB-2 (test) (j) CUB-3 (test)

Figure 6: Results on simulated learners. The x-axis is the number of teaching images; the y-axis is error probability.



(a) JM (test) (b) VW (test) (c) CUB-1 (test) (d) CUB-2 (test) (e) CUB-3 (test)

Figure 7: Results on real human learners. The vertical axis displays the accuracy of human learners at test time.

Real human learners. We conducted five user studies on Amazon Mechanical Turk (MTurk). Participants were randomly assigned to one of three different teaching strategies, where each strategy received on the order of 30 participants. They were shown 10, 12, and 5 teaching and 12 testing images for the JM, VW, CUB datasets respectively. In order to motivate the participants, we paid a one dollar bonus to the top 10% of participants based on their performance at test time. Experiments were conducted using the same protocol as Johns et al. (2015), where learners were shown a sequence of images during the "teaching" phase and after each image they were asked to estimate the correct class label. After estimating the class label for a teaching image, they were given the ground truth class label as feedback. In the case of NOTES, they were also shown an additional 4, 1, and 2 interpretable features for the JM, VW, CUB datasets - displayed as an arrow pointing to the feature locations in each image. Teaching was then followed by a "test" phase, similar to teaching, where there was no feedback after the learners' responses. We used the same randomly selected set of test images to compare all methods.

Results for real human workers are shown in Fig. 7. Overall the performance improvement is not as pronounced as with the simulated learners. This can partially be attributed to the inherent noise of MTurk participants and a potential mismatch between the learner and teacher hypothesis spaces. Nevertheless, NOTES still finishes within the top two on average for each dataset. One interesting limitation is the VW dataset, where the learners found it difficult to notice small size differences in the objects.

7 Conclusion

We presented NOTES, a principled approach for constructing interpretable teaching sets that uses simple, yet informative, explanations to teach novel concepts to learners. We prove that our approach is competitive with the optimal explanation-based teaching strategy. In addition, through experiments on both simulated and real humans, we show that learners taught with interpretable explanations outperform those taught with weaker class label information. Interpretable teaching algorithms like NOTES offer the potential to automatically and efficiently teach a diverse set of topics to large numbers of human learners.

Acknowledgments. This work was supported in part by Northrop Grumman, Bloomberg, AWS Research Credits, Google as part of the Visipedia project, and a Swiss NSF Early Mobility Postdoctoral Fellowship.

References

Ba, Jimmy and Caruana, Rich. Do deep nets really need to be deep? In *NIPS*, 2014.

Bak, Ji Hyun, Choi, Jung Yoon, Akrami, Athena, Witten, Ilana, and Pillow, Jonathan W. Adaptive optimal training of animal behavior. In *NIPS*, 2016.

Basu, Sumit and Christensen, Janara. Teaching classification boundaries to humans. In *AAAI*, 2013.

Cakmak, Maya and Thomaz, Andrea L. Eliciting good teaching from humans for machine learners. *Artificial Intelligence*, 217:198–215, 2014.

Canfield, Ward. Aleks: A web-based intelligent tutoring system. *Mathematics and Computer Education*, 2001.

Chang, Jonathan, Gerrish, Sean, Wang, Chong, Boyd-Graber, Jordan L, and Blei, David M. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

Chen, Xi, Chen, Xi, Cheng, Yu, and Tang, Bo. On the recursive teaching dimension of vc classes. In *NIPS*, 2016.

Corbett, Albert T and Anderson, John R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 1994.

Cover, Thomas and Hart, Peter. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13 (1):21–27, 1967.

Eaves, Baxter, Schweinhart, April, and Shafto, Patrick. Tractable bayesian teaching. In *Big Data in Cognitive Science*, 10 2015.

Goldman, Sally A and Kearns, Michael J. On the complexity of teaching. *JCSS*, 1995.

Grant, Elizabeth R and Spivey, Michael J. Eye movements and problem solving: Guiding attention guides thought. *Psychological Science*, 2003.

Johns, Edward, Mac Aodha, Oisin, and Brostow, Gabriel J. Becoming the expert-interactive multi-class machine teaching. In *CVPR*, 2015.

Koedinger, Kenneth R, Anderson, John R, Hadley, William H, and Mark, Mary A. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8:30–43, 1997.

Lakkaraju, Himabindu, Bach, Stephen H, and Leskovec, Jure. Interpretable decision sets: A joint framework for description and prediction. In *KDD*, 2016.

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

Liu, Weiyang, Dai, Bo, Rehg, James M, and Song, Le. Iterative machine teaching. In *ICML*, 2017.

Mac Aodha, Oisin, Su, Shihan, Chen, Yuxin, Perona, Pietro, and Yue, Yisong. Teaching categories to human learners with visual explanations. In *CVPR*, 2018.

Nemhauser, George L, Wolsey, Laurence A, and Fisher, Marshall L. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.

Patil, Kaustubh R, Zhu, Xiaojin, Kopeć, Łukasz, and Love, Bradley C. Optimal teaching for limited-capacity human learners. In *NIPS*, 2014.

Piech, Chris, Bassen, Jonathan, Huang, Jonathan, Ganguli, Surya, Sahami, Mehran, Guibas, Leonidas J, and Sohl-Dickstein, Jascha. Deep knowledge tracing. In *NIPS*, 2015.

Rafferty, Anna N, Brunskill, Emma, Griffiths, Thomas L, and Shafto, Patrick. Faster teaching by pomdp planning. In *AIED*, 2011.

Ribeiro, Marco Tulio, Singh, Sameer, and Guestrin, Carlos. Why should i trust you?: Explaining the predictions of any classifier. In *KDD*, 2016.

Roads, Brett, Mozer, Michael C, and Busey, Thomas A. Using highlighting to train attentional expertise. *PloS one*, 2016.

Romero, Adriana, Ballas, Nicolas, Kahou, Samira Ebrahimi, Chassang, Antoine, Gatta, Carlo, and Bengio, Yoshua. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Singla, Adish, Bogunovic, Ilija, Bartók, Gábor, Karbasi, Amin, and Krause, Andreas. Near-optimally teaching the crowd to classify. In *ICML*, 2014.

Sullivan, Brian L, Wood, Christopher L, Iliff, Marshall J, Bonney, Rick E, Fink, Daniel, and Kelling, Steve. ebird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 2009.

Ustun, Berk and Rudin, Cynthia. Methods and models for interpretable linear classification. *arXiv preprint arXiv:1405.4047*, 2014.

von Ahn, Luis. Duolingo: learn a language for free while helping to translate the web. In *IUI*, 2013.

Wah, Catherine, Branson, Steve, Welinder, Peter, Perona, Pietro, and Belongie, Serge. The Caltech-UCSD birds-200-2011 dataset, 2011.

Whitehill, Jacob and Movellan, Javier. Approximately optimal teaching of approximately optimal learners. *Transactions on Learning Technologies*, 2017.

Zhu, Xiaojin. Machine teaching for bayesian learners in the exponential family. In *NIPS*, 2013.

8 Proofs

Proof of Proposition 1. One can easily construct problem instances where the label-based strategy requires an arbitrary number of teaching examples (up to the size of the full teaching set) compared to the explanation based strategy. In an extreme case, consider the example where an explanation reveals the single informative feature in the data, while there are infinitely many dummy features that are not correlated with the label. An explanation-based teaching strategy explicitly points out the informative feature to the learner (in other words, the teacher shows an explanation which directly leads to the optimal (constant) attention function f^*); while label-based strategy requires the learner to discover the feature among an infinite pool of features. In the worst-case scenario, the label-based learner cannot uncover the true hypothesis unless the teacher reveals the labels of all examples. However, with explanation-based teaching one may instantly discover the hypothesis based on the informative feature. □

Proof of Lemma 2. We now show that our objective function $r(\emptyset)-r(\mathcal{S})$ is monotone submodular. First, it is clear that the sum of the weight of the edges will only be discounted when we show more examples with their corresponding labels and explanations. For any edge, we never increase its weight, and therefore the remaining weight $r(\mathcal{S})$ is monotone non-increasing.

To prove submodularity, notice that the reduction of weight for each individual edge $\{f,g\}\in\mathcal{E}$ is a submodular function. The reduction of total weight is a sum of submodular functions and hence is also submodular. □

Proof of Lemma 3. By definition, the error rate of a hypothesis h represents the probability that h makes an error on $\mathbf{x}$ drawn uniformly from $\mathcal{X}$, i.e., $\mathrm{err}(h)=\mathbb{P}[\mathbf{x}\notin\{\mathbf{x}_i : h(\mathbf{x}_i)=y_i\}]$. Suppose that hypothesis h is a composition of f and g, i.e., $h=g\circ f$. We know that h will make the correct prediction on $\mathbf{x}$, as long as both f and g make the correct predictions on $\mathbf{x}$. Denote the set of examples on which h makes the correct prediction as $\mathcal{X}_h:=\{\mathbf{x}_i : h(\mathbf{x}_i)=y_i\}$; similarly, define $\mathcal{X}_f:=\{\mathbf{x}: f(\mathbf{x}_i)=\mathbf{e}_i\}$, and $\mathcal{X}_g:=\{\mathbf{x}: g(\phi(\mathbf{x}_i),\mathbf{e}_i)=y_i\}$. From the above discussion, we get $\mathcal{X}_f\cap\mathcal{X}_g\subseteq\mathcal{X}_h$. Therefore,

$$\mathcal{X}_h^{\complement}\subseteq(\mathcal{X}_f\cap\mathcal{X}_g)^{\complement}={\mathcal{X}_f}^{\complement}\cup{\mathcal{X}_g}^{\complement}, \tag{8.1}$$

where $\mathcal{X}_h^{\complement}=\mathcal{X}\setminus\mathcal{X}_h$ represents the set complement. Note that $\mathrm{err}(h)=\frac{|\{i\in\{1,\dots,m\}:h(\mathbf{x}_i)\neq y_i\}|}{m}=\mathbb{P}\left[\mathbf{x}\in{\mathcal{X}_h}^{\complement}\right]$, $\mathrm{err}(f)=\frac{|\{\{i\in\{1,\dots,m\}:f(\mathbf{x}_i)\neq\mathbf{e}_i\}|}{m}=\mathbb{P}\left[\mathbf{x}\in{\mathcal{X}_f}^{\complement}\right]$ and $\mathrm{err}(g)=\frac{|\{\{i\in\{1,\dots,m\}:g(\phi(\mathbf{x}_i),\mathbf{e}_i)\neq y_i\}|}{m}=\mathbb{P}\left[\mathbf{x}\in{\mathcal{X}_g}^{\complement}\right]$. Therefore,

$$\begin{aligned}\mathrm{err}(h)=\mathbb{P}\left[\mathbf{x}\in{\mathcal{X}_h}^{\complement}\right]&\overset{(a)}{\leq}\mathbb{P}\left[\mathbf{x}\in{\mathcal{X}_f}^{\complement}\cup{\mathcal{X}_g}^{\complement}\right]\\&\overset{(b)}{\leq}\mathbb{P}\left[{\mathcal{X}_f}^{\complement}\right]+\mathbb{P}\left[{\mathcal{X}_g}^{\complement}\right]=\mathrm{err}(f)+\mathrm{err}(g).\end{aligned} \tag{8.2}$$

where step (a) is by Eq. (8.1) and step (b) holds by the union bound.

Recall from Eq. (4.2), our surrogate objective is defined as

$$\begin{aligned}r(\mathcal{S})&=\sum_{\{f,g\}\in\mathcal{E}}w(\{f,g\})\prod_{i\in\mathcal{S}}\mathbb{P}[\mathbf{e}_i,y_i\,|\,f,g]\\&=\sum_{f=f^*,g}w((f^*,g))\prod_{i\in\mathcal{S}}\mathbb{P}[\mathbf{e}_i,y_i\,|\,f^*,g]+\sum_{f,g=g^*}w((f,g^*))\prod_{i\in\mathcal{S}}\mathbb{P}[\mathbf{e}_i,y_i\,|\,f,g^*]\\&\overset{\text{Eq. (4.1)}}{=}\sum_{f=f^*,g}\mathbb{P}[f^*]\mathbb{P}[g]\mathrm{err}(g\circ f^*)\prod_{i\in\mathcal{S}}\mathbb{P}[\mathbf{e}_i\,|\,f^*]\mathbb{P}[y_i\,|\,\mathbf{e}_i,g]+\sum_{f,g=g^*}\mathbb{P}[f]\mathbb{P}[g^*]\mathrm{err}(g^*\circ f)\prod_{i\in\mathcal{S}}\mathbb{P}[\mathbf{e}_i\,|\,f]\mathbb{P}[y_i\,|\,\mathbf{e}_i,g^*]\\&=\mathbb{P}[f^*]\sum_{g}\mathbb{P}[g]\mathrm{err}(g)\prod_{i\in\mathcal{S}}\mathbb{P}[y_i\,|\,\mathbf{e}_i,g]+\mathbb{P}[g^*]\sum_{f}\mathbb{P}[f]\mathrm{err}(f)\prod_{i\in\mathcal{S}}\mathbb{P}[\mathbf{e}_i\,|\,f].\end{aligned} \tag{8.3}$$

For notation simplicity, let us define

$$\mathbb{Q}[f\,|\,\mathcal{S}]=\mathbb{P}[f]\prod_{i\in\mathcal{S}}\mathbb{P}[\mathbf{e}_i\,|\,f]$$

$$\mathbb{Q}[g\,|\,\mathcal{S}]=\mathbb{P}[g]\prod_{i\in\mathcal{S}}\mathbb{P}[y_i\,|\,\mathbf{e}_i,g]$$

as the unnormalized posterior probability of the attention function and decision function, respectively. Taking an expectation over h (and hence over f, g) on both sides of Eq. (8.2) we get an upper bound of $\mathbb{E}[\mathrm{err}(h)\,|\,\mathcal{S}]$:

$$\begin{aligned}\mathbb{E}[\mathrm{err}(h)\,|\,\mathcal{S}] &\leq \mathbb{E}[\mathrm{err}(f)+\mathrm{err}(g)\,|\,\mathcal{S}]\\ &= \sum_f \mathbb{P}[f\,|\,\mathcal{S}]\mathrm{err}(f)+\sum_g \mathbb{P}[g\,|\,\mathcal{S}]\mathrm{err}(g)\\ &\leq \sum_f \frac{\mathbb{P}[f\,|\,\mathcal{S}]}{\mathbb{P}[f^*\,|\,\mathcal{S}]}\mathrm{err}(f)+\sum_g \frac{\mathbb{P}[g\,|\,\mathcal{S}]}{\mathbb{P}[g^*\,|\,\mathcal{S}]}\mathrm{err}(g)\\ &\overset{(a)}{=} \sum_f \frac{\mathbb{Q}[f\,|\,\mathcal{S}]}{\mathbb{P}[f^*]}\mathrm{err}(f)+\sum_g \frac{\mathbb{Q}[g\,|\,\mathcal{S}]}{\mathbb{P}[g^*]}\mathrm{err}(g)\\ &= \frac{1}{\mathbb{P}[f^*]\mathbb{P}[g^*]}\left\{\sum_f \mathbb{P}[g^*]\mathbb{Q}[f\,|\,\mathcal{S}]\mathrm{err}(f)+\sum_g \mathbb{P}[f^*]\mathbb{Q}[g\,|\,\mathcal{S}]\mathrm{err}(g)\right\}\\ &= \frac{1}{\mathbb{P}[f^*]\mathbb{P}[g^*]}r(\mathcal{S})\end{aligned}$$

which gives us the lower bound on $r(\mathcal{S})$. Step (a) in the above equations is due to the fact that

$$\frac{\mathbb{P}[f\,|\,\mathcal{S}]}{\mathbb{P}[f^*\,|\,\mathcal{S}]}=\frac{\mathbb{Q}[f\,|\,\mathcal{S}]}{\mathbb{Q}[f^*\,|\,\mathcal{S}]}=\frac{\mathbb{Q}[f\,|\,\mathcal{S}]}{\mathbb{P}[f^*]},\text{ and }\frac{\mathbb{P}[g\,|\,\mathcal{S}]}{\mathbb{P}[g^*\,|\,\mathcal{S}]}=\frac{\mathbb{Q}[g\,|\,\mathcal{S}]}{\mathbb{Q}[g^*\,|\,\mathcal{S}]}=\frac{\mathbb{Q}[g\,|\,\mathcal{S}]}{\mathbb{P}[g^*]}.$$

Here, $\mathbb{Q}[f^*\,|\,\mathcal{S}]=\mathbb{P}[f^*]$ and $\mathbb{Q}[g^*\,|\,\mathcal{S}]=\mathbb{P}[g^*]$ follow from the definition of $\mathbb{Q}[\cdot\,|\,\mathcal{S}]$ and the fact that the probabilities of f^* and g^* are never discounted.

For the lower bound of $\mathrm{err}(h)$, observe that

$$\begin{aligned}\mathbb{E}[\mathrm{err}(h)\,|\,\mathcal{S}] &= \sum_h \mathbb{P}[h\,|\,\mathcal{S}]\mathrm{err}(h)\\ &= \sum_g\sum_f \mathbb{P}[g\circ f\,|\,\mathcal{S}]\mathrm{err}(g\circ f)\\ &\geq \sum_f \mathbb{P}[g^*\circ f\,|\,\mathcal{S}]\mathrm{err}(g^*\circ f)+\sum_g \mathbb{P}[g\circ f^*\,|\,\mathcal{S}]\mathrm{err}(g\circ f^*)\\ &= \sum_f \mathbb{P}[g^*\,|\,\mathcal{S}]\mathbb{P}[f\,|\,\mathcal{S}]\mathrm{err}(f)+\sum_g \mathbb{P}[f^*\,|\,\mathcal{S}]\mathbb{P}[g\,|\,\mathcal{S}]\mathrm{err}(g)\\ &\geq \sum_f \mathbb{P}[g^*]\mathbb{P}[f\,|\,\mathcal{S}]\mathrm{err}(f)+\sum_g \mathbb{P}[f^*]\mathbb{P}[g\,|\,\mathcal{S}]\mathrm{err}(g)\\ &= r(\mathcal{S})\end{aligned}$$

which completes the proof. □

Proof of Theorem 4. By the seminal results of Nemhauser et al. (1978), we know that for maximizing a monotone submodular function f under cardinality constraint, it holds that the value of the greedily selected set satisfies

$$f(\mathcal{S}_\ell)\geq\left(1-e^{\ell/k}\right)\max_{S:|S|\leq k} f(S).$$

Let k^* be the cardinality of the smallest set, such that $r(\mathcal{S}) \leq \mathbb{P}\,[f^*]\;\mathbb{P}\,[g^*]\;\epsilon/2$. Therefore, $\min_{\mathcal{S}:|\mathcal{S}|=k^*} r(\mathcal{S}) \leq \mathbb{P}\,[f^*]\,\mathbb{P}\,[g^*]\,\epsilon/2$. Let $\ell = k^*\log\frac{2W}{\mathbb{P}[f^*]\mathbb{P}[g^*]\epsilon}$, where $W = r(\emptyset)$ denotes the initial weight of the bipartite graph. Let $\mathcal{S}_\ell$ be the remaining weight after showing ℓ teaching examples. We get

$$\begin{aligned}r(\emptyset)-r(\mathcal{S}_\ell) &\geq \left(1-e^{\ell/k^*}\right)(r(\emptyset)-\mathbb{P}[f^*]\mathbb{P}[g^*]\epsilon/2)=\left(1-\frac{\mathbb{P}[f^*]\mathbb{P}[g^*]\epsilon}{2E}\right)\left(r(\emptyset)-\frac{\mathbb{P}[f^*]\mathbb{P}[g^*]\epsilon}{2}\right)\\ &\geq r(\emptyset)-\mathbb{P}[f^*]\mathbb{P}[g^*]\epsilon,\end{aligned}$$

which proves the result. □