The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

Teaching to Learn: Sequential Teaching of Learners with Internal States

Mustafa Mert Çelikok[1], **Pierre-Alexandre Murena**[1], **Samuel Kaski**[1, 2]

[1] Aalto University
[2] The University of Manchester
mustafamert.celikok@aalto.fi, pierre-alexandre.murena@aalto.fi, samuel.kaski@aalto.fi

Abstract

In sequential machine teaching, a teacher's objective is to provide the optimal sequence of inputs to sequential learners in order to guide them towards the best model. However, this teaching objective considers a restricted class of learners with fixed inductive biases. In this paper, we extend the machine teaching framework to learners that can improve their inductive biases, represented as latent internal states, in order to generalize to new datasets. We introduce a novel framework in which learners' inductive biases may change with the teaching interaction, which affects the learning performance in future tasks. In order to teach such learners, we propose a multi-objective control approach that takes the future performance of the learner after teaching into account. This framework provides tools for modelling learners with internal states, humans and meta-learning algorithms alike. Furthermore, we distinguish manipulative teaching, which can be done by effectively hiding data and also used for indoctrination, from teaching to learn which aims to help the learner become better at learning from new datasets in the absence of a teacher. Our empirical results demonstrate that our framework is able to reduce the number of required tasks for online meta-learning, and increases independent learning performance of simulated human users in future tasks.

Introduction

Pedagogical systems can be described as intelligent systems in which an agent, called the *teacher*, transmits information to a second agent, called the *learner* in order to help them learn a target concept (Shafto, Goodman, and Griffiths 2014). *Machine teaching* has emerged as a computational model for pedagogical systems, which addresses the problem of finding the best training data that can guide a learner, human or machine alike (Patil et al. 2014; Chen et al. 2018), to a target model with minimal effort (Zhu 2015; Goldman and Kearns 1995). However, conventional machine teaching considers a restricted class of learners which have fixed inductive biases (e.g. parameter initialization, model family, network architecture, variable selection etc.) and hyperparameters. Learners in this class are not able to update their inductive biases during the learning process, which excludes machine learning methods such as meta-learning and human learners who can *learn to learn*, to achieve better generalization amongst similar learning tasks (Griffiths et al. 2019). The optimal strategies for teaching learners with fixed inductive biases differ from the strategies for learners who are able to learn better inductive biases.

The mathematical framework we present in this paper shows that if a learner's initial biases are unsuitable for the current task and cannot change, the teacher may have no choice but to hide parts of the training data from the learner in order to teach them a good model. More specifically, the learner can be taught a better model by hiding parts of the data than by providing the full dataset. For teaching machine learning algorithms, this teaching strategy is close to data-poisoning (Mei and Zhu 2015), and for teaching humans, it is undesirable behaviour which attempts to manipulate the learner. However, considering that the learner's biases can change and be influenced by the teacher induces a completely different teaching strategy: helping the learner refine their *internal state*, essentially teaching them better inductive biases. This empowers the learners by teaching them to perform better during the learning phase, with assistance of the teacher, but also in future tasks, even in the absence of a teacher. We refer to this more advanced teaching strategy as *Teaching to Learn (TtL)*.

In this paper, we present a mathematical framework for TtL. We formalize this setting as a two-player game involving a *learner* and a *teacher*, and take the perspective of the teacher. We unify the definition of inductive biases in machine learning algorithms (e.g. model family, initialization etc.) and in humans (prior task knowledge, meta-knowledge, etc.), and model both cases as a latent *internal state* of the learner. A key difference of the new framework from conventional machine teaching is that the internal state of the learner changes over time as a result of the teacher's actions. Thus, in the new framework, the task of the teacher consists not only of guiding the learner towards a model close enough to the best model possible, but also of guaranteeing that the learner will be able to learn good models without supervision in future similar tasks. To do so, the teacher needs to lead the learner to an internal state which consists of inductive biases that are suitable for current and future tasks.

The main contributions are: (i) We generalize sequential machine teaching to a setting where the learner has an internal state which affects their preferences over models, and evolves over time in response to the teacher's actions. Unlike conventional machine teaching, the new generalization leads to a multi-objective formulation of teaching goals. (ii) We show that when the learner's internal state is static and suboptimal, optimal teaching is possible only at the price of data manipulation, defined in detail below. (iii) We show that augmenting machine teaching by considering the teacher's influence on the learner's internal states allows the teacher to avoid manipulative strategies and help the learner learn to perform better later, even in the absence of the teacher. Our work bridges a gap between machine teaching and automated human teaching, and proposes a mathematical framework that can bring the two closer together.

An Illustrative Example

Consider an explorative data analysis setting where an AI assistant is helping its users build linear models such as $\mathbf{Y} = \mathbf{X}\xi + \epsilon$ with $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$, for their data $\mathcal{D} = \{(X_i, Y_i)\}_{i=1\dots n}$, where $X_i \in \mathbb{R}^d$ and $Y_i \in \mathbb{R}$. In particular, the AI assists the human user in the selection of covariates, to discover linear relationships between inputs $\mathbf{X}$ and outputs $\mathbf{Y}$. Here, the end goal is not to just get a good predictive model, but also for the human to uncover important covariates and their relationship to the output. If the AI knows the human is an expert statistician, it can automatically compute which covariates are important, present all the facts to the human, and be confident that the user will draw the right conclusions. In this case, no pedagogical behaviour is needed from the AI. However, if for instance the user does not know what collinearity means, they may wrongly think that collinear variables with strong output correlation must all be included in the regression. On the contrary, when one of the collinear variables is included, the rest will not improve the regression, and their inclusion would result in unidentifiable regression weights. In this case, the AI cannot assist effectively, since its recommendations for which covariates to include/exclude may get rejected. This issue cannot be resolved easily with explanations or data visualization, since understanding the explanations requires the conceptual knowledge of collinearity. If the AI assistant can teach the user about collinearity during the interaction, this will improve things for both the current and future model building tasks. However, when there are many such concepts as collinearity involved, the AI cannot simply present tutorial material on everything to a user. Thus, the AI assistant must infer what the user knows and may not know, and try to tutor only when necessary.

In the scenario above, the AI assistant can model its user as a learning algorithm which, given data as input, produces a linear model as output. The task of the assistant is to help the learning algorithm converge to a good model with minimal effort. Evidently, this setting can be modelled as machine teaching where the system is the teacher, and the user is the learner. If we apply conventional machine teaching here by treating covariate suggestions as data, the teacher's optimal behaviour would be to avoid suggesting collinear covariates. Such a strategy is optimal in terms of the preexisting optimality criteria in machine teaching, since it prevents the user from including collinear covariates. However, the model would then be built by effectively hiding from the user information that they could misinterpret: Had the user observed the entire dataset by themself, they would have included collinear covariates and ended up with a different model. This is not satisfying since it implies they will not be able to choose a good model for future datasets, unless the teacher is there to supervise them. Moreover, this discrepancy between the model built with and without the supervision of the teacher can be interpreted as resulting from a manipulative teaching strategy. Manipulation is particularly undesirable in the case of human learners.

The situation above can be avoided by allowing the teacher to influence the modelling biases and preferences of the learner, corresponding to their internal state. If a teacher can infer that the learner's modelling preferences do not depend on collinearity, it can follow an alternative strategy that helps the learner understand the notion of collinearity, and therefore changes their internal state for the better. However, the current optimality criteria used in machine teaching are based only on the final model obtained, thus a new criterion is needed to learn such teaching strategies.

The mathematical framework we present in this paper formalizes the intuitions described above. In particular, we will demonstrate in Proposition 1 below that unless the teaching aims at changing the learner's internal state, the teacher's choices are either to manipulate the data seen by the learner or end up with suboptimal learning results. A crucial insight of our work is that, if the teacher takes the learner's future modelling performance in the teacher's absence into account, the induced optimal teaching strategy can lead to beneficial changes in the learner's internal state, utilizing whatever actions are available. The optimality of such a teaching policy will be exposed in Proposition 2.

Teaching to Learn (TtL)

In this section, we formalize the intuitions discussed above and present the mathematical framework of Teaching to Learn.

Sequential Teaching of Models

Learning task. The learning task of a learner is defined as inferring a model $\theta \in \Theta$ to describe a dataset $\mathcal{D}$, where Θ denotes the model space. We define a discrepancy $d : \Theta \times \Theta \mapsto [0, \infty)$ as a function that satisfies the property $d(\theta_1, \theta_2) = 0 \iff \theta_1 = \theta_2$. For instance, in the case of probabilistic modelling, if θ is the posterior distribution over model parameters then d is a discrepancy measure between probability distributions such as KL-divergence.

The multi-agent model of teacher—learner interaction. We consider two agents: a learner and a teacher. The teacher has better inductive biases than the learner, and therefore can identify a better model $\theta^* \in \Theta$. The learner aims to select a model to describe the data. The interaction between the two agents is modelled as a sequential game. At each time step t, the teacher selects an action $a_t \in \mathcal{A}$ to perform, the learner responds with an action $b_t \in \mathcal{B}$ and updates its selected model θ_t. Every action of the teacher a_t can be considered as suggesting a model or a hypothesis to the learner.

The learner may accept or reject this suggestion, or simply ignore it when updating its model. We do not have direct control over the learner, thus we take the perspective of the teacher whose goal is to identify the optimal sequence of actions minimizing $d(\theta_T, \theta^*)$ for a certain horizon T.

Learner's internal states. The learner's internal state space $\mathcal{Z}$ is represented as the product of a function space $\mathcal{F}$ and a set of learning algorithms Π. An internal state is then an element of $\mathcal{Z}$, defined as the tuple $z = (f, Alg(D; f, \Theta))$, where f is a real-valued function inducing a preference ordering in Θ for the data D. The f models the learner's inductive biases and prior knowledge as an a priori preference over models. The $Alg(D; f, \Theta)$ denotes the learning algorithm the learner uses to build a model of the dataset D, parameterized by the model space Θ and modelling preferences. The definition of Alg is general: it can be stochastic or deterministic, can output a single model, distribution over models, or the average model. A learner's state is the tuple $s_\ell = (z, \theta)$, thus $\mathcal{S}_\ell = \mathcal{Z} \times \Theta$. The probability of a learner with internal state z responding to the teacher action a_t with b_t is denoted as $\pi_\ell(b_t | a_t, s_{\ell,t} = (z_t, \theta_t))$. We refer to the supplementary material [1] for an in-depth discussion and guidance on the design of $\mathcal{F}$ and Π. We model the evolution of the learner's internal state with transition probabilities $p(z_{t+1} | s_{\ell,t}, a_t, b_t)$, which induce an *internal state dynamics*. In many cases, the z_t evolves conditionally independently of the θ_t, since the two are only coupled by the data. Thus, in our experiments, we will assume the factorization $p(z_{t+1} | s_{\ell,t}, a_t, b_t) = p(z_{t+1} | z_t, a_t, b_t)$. For ease of exposure, we will group any teacher action a that can influence a learner's internal state under an action called *tutor*. This is not a modelling restriction per se, but simply a placeholder for a simpler exposure of our model.

Teacher's decision-making task. We model the teacher's decision-making for teaching a learner with learner's policy π_ℓ as a POMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O})$, where $\mathcal{S} = \Theta \times \mathcal{Z}$ is the state space, $\mathcal{A}$ the space of actions introduced earlier, $\mathcal{T}$ the transition kernel, Ω the set of observations, $\mathcal{O}$ a set of conditional observation probabilities and $\mathcal{R}$ a cost function that will be discussed in the conclusion of the next section. A state $s \in \mathcal{S}$ is composed of two components $s = (\theta, z)$, where θ is the model selected by the learner and z is the learner's internal state. The θ is fully observable, but z cannot be directly observed by the teacher. However, it can be inferred from the learner's policy $\pi_\ell(b_t | a_t, s)$, therefore $\Omega = \mathcal{B}$ and $\mathcal{O} = \pi_\ell$. The transition dynamics $\mathcal{T}(s' \mid s, a)$ is induced by the learner's algorithm Alg which governs how the θ component evolves, and its internal state dynamics. Fixing a teacher policy π_τ induces the state chain $p^{\pi_\tau}(s_{t+1} | s_t)$.

The Non-manipulative Teaching Objective

As described in the illustrative example, we would like the teacher to avoid manipulating the data the learner gets to observe. We formalize this notion as follows: Manipulation level measures the discrepancy between the model learned by a learner by interacting with a teacher and the model that the learner would infer from the whole dataset, without assistance from a teacher. Intuitively, manipulation level measures how much the teacher has influenced the learning outcome.

Definition 1 (Manipulation). *Given a learner's internal state $z = (f, Alg)$, let $\theta_\ell = Alg(\mathcal{D}; f, \Theta)$ denote the model the learner infers from the dataset $\mathcal{D}$ without a teacher. The* ***manipulation level*** *of a teacher-learner interaction for the same task is given by $Manip(z, \mathcal{D}, \theta_\tau) = d(\theta_\ell, \theta_\tau)$ where the θ_τ is the model learner infers at the end of teaching.*

For some internal states, achieving zero manipulation can be possible. If the learner reaches these states, then it does not need a teacher to manipulate the data it receives. Trivially, if Alg ignores data, the learner guarantees zero manipulation. We will exclude such internal states, as they are unrealistic.

Definition 2 (Enlightened Internal State). *Inspired by Immanuel Kant's definition of enlightenment, we say that an internal state $z \in \mathcal{Z}$ is* ***enlightened*** *for dataset $\mathcal{D}$ toward model θ if $Manip(z, \mathcal{D}, \theta) = 0$.*

In the following propositions (proofs in supplementary material), we demonstrate the importance of modelling the internal state transitions to provide optimal and non-manipulative teaching. We denote the set of all enlightened internal states for data $\mathcal{D}$ towards the model θ by $\mathcal{Z}^*(\theta, \mathcal{D}) \subset \mathcal{Z}$. We will denote an objectively good (or optimal) model for the dataset $\mathcal{D}$ with θ^*.

Proposition 1. *Suppose that the initial internal state of the learner is $z_0 \notin \mathcal{Z}^*(\theta^*, \mathcal{D})$, and that for all $t > 0$, $p^{\pi_\tau}(z_t \in \mathcal{Z}^* | z_0) = 0$. Let θ be chosen by the learner at time t during a teaching process, such that $\theta_t = \theta$. Then for any $t > 0$, with probability 1 at least one of the two following statements is true: (1) $Manip(z_t, \mathcal{D}, \theta^*) > 0$ or (2) There exists a model θ' such that $d(\theta', \theta^*) < d(\theta, \theta^*)$ and $p(\theta_t = \theta' | z_0) > 0$.*

Proposition 1 states that a teacher who would not enlighten the learner (e.g. by not triggering any change in learner's internal state) is necessarily limited to either being manipulative or being sub-optimal. This impossibility result applies in particular to machine teaching techniques which allow the teachers to alter the data distribution by filtering out samples or providing data that is inconsistent with the data distribution as shown by Peltola et al. (2019).

The following proposition states that, when internal states can be influenced by the teacher, the teacher can guide the learner towards an internal state where θ^* could be inferred without assistance.

Proposition 2. *If there exists an enlightened internal state z^* that is reachable with probability p^* from the initial state z_0 under a teacher policy π_τ, then there exists a teacher policy π such that, with probability of at least p^*, the θ_T obtained by teaching interaction is optimal ($\theta_T = \theta^*$) and the teaching is non-manipulative ($Manip(z_T, \mathcal{D}, \theta) = 0$).*

These two propositions imply that optimal teaching can be made non-manipulative by allowing the teacher to help the learner switch to an enlightened internal state. Here, non-manipulative teaching means that the learner is eventually

[1] Supplementary material available at arXiv:2009.06227

able to make the same choice of a model without any supervision. Another desirable property of learning is the ability for the learner to perform correctly on new datasets.

Corollary 1. *Let $(\mathcal{D},\theta^*)$ and $(\mathcal{D}',\theta'^*)$ be two learning tasks with datasets and associated target models, and suppose that $z^* \in \mathcal{Z}^*(\theta'^*,\mathcal{D}') \cap \mathcal{Z}^*(\theta^*,\mathcal{D})$. Then $Alg_{z^*}(\mathcal{D}'; f_{z^*},\Theta) = \theta'^*$ and $Alg_{z^*}(\mathcal{D}; f_{z^*},\Theta) = \theta^*$.*

Corollary 1 highlights the inherent connection between our framework and meta-learning, also commonly referred to as *learning to learn* (Thrun and Pratt 2012; Vanschoren 2019). Indeed, a meta-learner aims to learn good inductive biases for a similar set of datasets. In our case, the inductive biases of a learner are represented by its internal states, and datasets that are similar for a learner have overlapping enlightened internal states.

A cost function for non-manipulative teaching. We have identified three desirable properties for a teaching policy in our framework: (O1) Assist the learner to select the optimal model θ^* for $\mathcal{D}$; (O2) Make the learner able to select the best model $\theta^* \in \Theta$ for $\mathcal{D}$ without assistance; (O3) Make the learner able to select the optimal model for tasks similar to $\mathcal{D}$ without assistance. The (O1) is the standard machine teaching objective, whereas (O2) captures the objective of enlightening the learner to make non-manipulative teaching optimal. It follows from corollary 1 that (O2) implies (O3). Using these insights, we propose a multi-objective cost function given by: (O1) the final model discrepancy $d(\theta_T,\theta^*)$; (O2) the final manipulation level $Manip(z,\mathcal{D},\theta_T)$; and (O3) model discrepancy for related tasks $\mathcal{D}'$: $\sum_{\mathcal{D}'} d(Alg_{z_T}(\mathcal{D}'; f_{z_t},\Theta),\theta^*(\mathcal{D}'))$. Since the $\mathcal{D}$ can be seen as a future task, (O3) includes the (O2). We use the linear scalarization method to map the multi-objective cost to a single objective function $g_T(z_T,\theta_T)$ with weights $u=(u_1,u_2)$ controlling which objective the teacher should prioritize.

$$g_T = u_1 d(\theta_T,\theta^*) + u_2 \sum_{\mathcal{D}'} d(Alg_{z_T}(\mathcal{D}'; f_{z_t},\Theta),\theta^*(\mathcal{D}')) \qquad (1)$$

Case I: Interactive Variable Selection with Users

We now apply our framework to the setup presented in the illustrative example, where the teacher helps a (simulated) user build linear models.

Description of the task. The goal of the learner is to choose which variables to include in the linear model. A variable can be excluded from the regression by setting its weight to zero as $\xi^i = 0$. Thus, the model space is the space of d-dimensional binary vectors $\Theta = \{0,1\}^d$ with each dimension denoted as $\theta^i = \mathbb{I}(\xi^i \neq 0)$ where $\mathbb{I}$ is the indicator function. At each time-step the teacher can select a variable $i \in \{1,\ldots,d\}$ from the dataset to display or provide explicit explanations about the design of linear models (which corresponds to an action called $tutor$). Therefore, the action space of the teacher is $\mathcal{A} = \{1,\ldots,d\} \cup \{tutor\}$. At time t, the learner observes action a_t from the teacher and picks a response $b_t \in \{0,1\}$ corresponding to rejecting or accepting the suggestion of the teacher. In case $a_t = i \in \{1,\ldots,d\}$ is not a tutoring action, the learner updates the model θ_t based on whether they accepted to include the suggested variable or not, therefore $\theta_t^i = b_t$.

Learner's internal state space. When making modelling decisions, different learners pay attention to different statistics in the data and the model, but to extents unknown to the teacher. Based on this observation, the teacher formulates the learner's modelling preferences as functions of the form $f(\phi(\theta,a);\mathbf{w}_z) = \mathbf{w}_z^T\phi(\theta,a)$ where $\phi(\theta,a)$ is an embedding of the statistics, for a model suggested by the teacher through action $a \in \{1,\ldots,d\}$. The $\mathbf{w}_z$ is an unknown weight vector capturing how much the learner pays attention to each statistic. Therefore, the space of preference functions $\mathcal{F}$ (introduced in Sequential Teaching of Models) is defined as set of linear functions from the embedding space to $\mathbb{R}$. Since the learner is doing linear regression, the space of algorithms Π consists of a single algorithm which performs the regression.

The feature map ϕ (embedding) encodes the quantities of interest to the learner, i.e. here the correlation of the shown variable to the output, and (maximal) correlation with already included variables as $\phi(a_t,\theta_{t-1}) = (|corr(a_t,Y)|, \max_{j:\theta^j_{t-1}\neq 0}|corr(a_t,j)|)$.

With this internal state space, the general policy of the learner is then given by

$$b_t|a_t,z_t \sim Bernoulli\left(\sigma(f_{z_t}(\phi(a_t,\theta_{t-1})))\right). \qquad (2)$$

As discussed in the illustrative example, two classes of behaviours can be observed depending on whether the learner knows collinearity. Formally, we observe that this corresponds to the decomposition of $\mathcal{Z}$ into two subspaces: $\mathcal{Z} = \mathcal{Z}^{(0)} \cup \mathcal{Z}^{(1)}$. The subspace $\mathcal{Z}^{(0)}$, associated to $\mathcal{F}^{(0)} = \{f : x \mapsto \mathbf{w}^T x : \mathbf{w} = (w_1, 0), w_1 \in \mathbb{R}\}$, describes the behaviour of learners who do not know collinearity, whereas $\mathcal{Z}^{(1)}$, associated to $\mathcal{F}^{(1)} = \{f : x \mapsto \mathbf{w}^T x : \mathbf{w} = (w_1,w_2), w_1 \in \mathbb{R}, w_2 < 0\}$, describe learners who understand collinearity and would avoid including collinear variables in a model since this would make the coefficients difficult to interpret in an exploratory data analysis task.

Learner's internal state dynamics. We consider the following model for the transitions between learner's states. The transitions of internal states can only be triggered with the action $a_t = tutor$, with probability η, resulting in the following dynamics: $p(z_{t+1} \in \mathcal{Z}^{(1)}|z_t \in \mathcal{Z}^{(0)}, a_t \neq tutor) = 0$, $p(z_{t+1} \in \mathcal{Z}^{(1)}|z_t \in \mathcal{Z}^{(0)}, a_t = tutor) = \eta$ and $p(z_{t+1} \in \mathcal{Z}^{(1)}|z_t \in \mathcal{Z}^{(1)}) = 1$. As a consequence, the data-generating process for feedback b_t is a Markov-switching model (Hamilton 1989). This model extends the well-known Bayesian knowledge tracing (BKT) (Corbett and Anderson 1994) by creating a hierarchy where internal states are treated similarly to BKT, while the behaviour of the student is treated like a learning algorithm

Teacher's cost. We define a stage cost function $g : \mathcal{A} \rightarrow [0,\infty)$, and take $g(a)$ constant for all $a \in \{1,\ldots,d\}$, but this can be generalized to variable-specific costs (e.g. if some

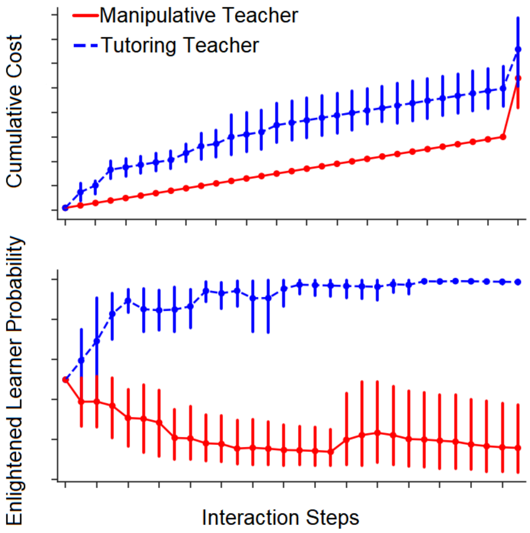

Figure 1: Comparison of a manipulative and a tutoring teacher (bars indicate 95% CI). Top: when only the performance on the current dataset matters for the terminal cost observed at the last time-step, the manipulative teaching (red) policy is cost-optimal and there is no need to tutor. Bottom: tutoring teacher (blue) leads to type changes from naive to enlightened.

features are more difficult to assess by the learner). We assume that the cost of the tutoring action $g(tutor)$ is higher than the cost of a variable recommendation. This is due to the fact that we expect tutoring actions to be intrusive to the task, and incur higher cognitive costs on the human learner. We complete the teaching with the terminal cost introduced in Equation 1.

Algorithm. In the POMDP with state $s = (\theta, z)$, the model θ is observed, but the learner's internal state z is not. It can be inferred with the posterior, $p(z_t|H_t)$, where the H_t denotes the interaction history. The detailed expression of this posterior is provided in the supplement. We solve this POMDP by using problem approximation (Bertsekas 2019) and turning this into a simpler fully-observed stochastic dynamic programming problem by repeating the following process: We take posterior expectations $\bar{\alpha}_t, \bar{\mathbf{w}}|H_t$ and sample from the space $\mathcal{Z}^{(n_t)}$ of the internal states $\tilde{n}_t \sim Bernoulli(\bar{\alpha}_k)$. We then use Monte Carlo rollouts by simulating the decision trajectory with a fixed parameter $\bar{w}$, based on the learner's state transition dynamics and policy given by Equations 2. We perform the first action a_{t+1} of the Monte Carlo planning solution. After getting learner's feedback b_{t+1}, the belief $p(\alpha_{t+1}, w|H_{t+1})$ is updated and the process is repeated.

Experimental Results

Setup. We use the data generation method provided by Ghosh and Ghattas (2015) for comparing method performances in collinear datasets, and generate random regression datasets with 10 independent and 15 collinear variables (details in the supplementary materials). Such high degree of collinearity is a typical feature of large-panel macroeconomic data (De Mol, Giannone, and Reichlin 2008). All results have been replicated with 10 random seeds and we present averaged values with 95% confidence intervals (CI). We simulate the learner's behaviour using the presented model (policy 2 and learner's internal state dynamics). Unless stated otherwise, the value for η is 0.5. Sensitivity analysis is in the supplement. The optimal variable selection strategy is to include all independent variables, and choose only one from the collinear variables. Once the variable selection is done, the learner pays a unit cost (1.0) for each missed independent variable and every extra collinear variable selected, which corresponds to a penalty $d(\theta, \theta^*)$, $d(.,.)$ being the Hamming distance

Experiment 1: Manipulative teaching is optimal for the current dataset. In standard iterative machine teaching, the goal is to guide the learner into the best possible model with minimal cost for a given dataset, which corresponds to the scalarization $u_1 = 1, u_2 = 0$ (only the current dataset is considered in the terminal cost). The cumulative cost in Figure 1 shows the performance of our rollout method (blue) against a teacher who is hard-coded to never choose the tutor action, which reduces to the standard iterative machine teaching (red). According to Proposition 1, such a teacher is expected to have a non-zero level of manipulation. Due to the stochasticity of the rollout approximation, our method chooses tutor action in multiple time-steps and thus has a higher cumulative cost. The optimal policy in this setting should never tutor, and can simply manipulate the learner by never showing more than one variable from a collinear group.

Experiment 2: Manipulative teaching leads to low performance in independent learning. In order to evaluate how the two types of learners (knows vs does not know collinearity) perform without the presence of a teacher, we generated 10 test datasets, having the same degree of collinearity as the sets used for teaching in Experiment 1. We observe that, on 10 datasets sampled from the task distribution, the enlightened learner gets a mean terminal cost of 2.18 (stdev 0.44), while the learner that does not know about collinearity gets 12.34 (stdev 0.29). As expected, in the absence of a teacher, the enlightened learner performs much better since it takes collinearity into account. This also highlights the importance of educating the learner for future learning tasks, even if it does not benefit the teaching of the current dataset.

Experiment 3: Including an estimate of independent learning performance to the cost leads to enlightenment. We generated a set of 10 additional datasets from the same generation process with the same degree of collinearity. Differently from test datasets, we use these to estimate the learner's independent performance in future tasks, i.e. the second term (with coefficient u_2 in equation 1) in the teacher's cost formulation. We set $u_1 = 0.5, u_2 = 0.5$, hence, the current and future performances are considered equally important. As seen in Figure 2, this makes the tutor-

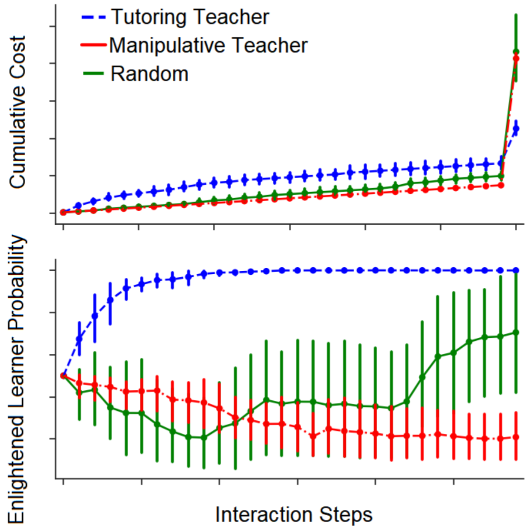

Figure 2: Comparison of mean teaching performances for manipulative, tutoring and random teachers with 95% CI. Top: TtL induces a lower cumulative cost than manipulative teaching, since an estimate of the learner's independent learning performance after interaction is included in the terminal cost observed at the last time-step. Bottom: TtL leads to a type change early on, whereas manipulative teaching does not cause any type changes.

ing teacher the best choice compared to the manipulative and random teachers: the cumulative cost of the tutoring teacher outperforms all, and the learner transitions to an enlightened internal state, as seen in our model's confident inference of the probability of the learner being enlightened. Since the learner becomes enlightened, its generalization performance improves drastically, as shown with Experiment 2. Details on how the tutoring teacher method induces internal state changes and how our model detects these changes in an episode are provided in the supplementary materials for two different values of η.

Case II: Teaching Online Meta-Learners

We next apply our framework to the case of teaching an online meta-learner to learn a good initialization. This section demonstrates that our framework is capable of bridging the gap between teaching humans and teaching machine learning algorithms.

Description of the task. Consider a learning task $\mathcal{T} \sim P(\mathcal{T})$ represented by a tuple $\mathcal{T} = (\mathcal{D}^{tr}, \mathcal{D}^{test})$ consisting of a training and a test dataset. All learning tasks that come from $P(\mathcal{T})$ have some common statistical properties. If a learner can exploit these common properties via inductive biases, it can generalize to new tasks faster. The goal of meta-learning is to learn these inductive biases from a set of tasks.

Model-agnostic meta-learning (MAML) (Finn, Abbeel, and Levine 2017) is a general framework for meta-learning applicable to any model that is trained by gradient descent. The goal of MAML for neural networks (NN) is to learn an initialization of the NN parameters θ_0 that quickly leads to good models for any task from $P(\mathcal{T})$. The initial model θ_0 can be seen as a form of modelling preferences and biases since the starting point on the parameter space indirectly induces a preference over the model space Θ due to finite data.

In order to learn a good θ_0, MAML uses a set of task samples $\{\mathcal{T}_i\}_{i=1,\dots,M}$ and minimizes the meta-learning loss $F(\theta) = \frac{1}{M}\sum_{i=1}^{M} \mathcal{L}(Alg(\mathcal{D}_i^{tr}, \theta), \mathcal{D}_i^{test})$, where $\theta \in \Theta$ corresponds to the parameters of the model. An online variant of this problem has been studied by Finn et al. (2019) where the meta-learner can get tasks only one by one.

In this section, we consider the new problem of teaching online meta-learners a good initialization θ_0^*.

Learner's internal state space and dynamics. For the type space of online meta-learners, the space of algorithms Π (see Sequential Teaching of Models) consists of a single learning algorithm Alg which is stochastic gradient descent. In this setting, the space of modelling preference functions $\mathcal{F}$ is implicit, yet we can assume $\mathcal{F}$ is parameterized by θ_0 since each initialization induces a preference. Thus, instead of $\mathcal{F}$ we will use Θ. The meta-learner always accepts proposed datasets ($\pi_\ell(b_t = 1) = 1$) and updates θ_0 by using the sublinear regret method introduced by Finn et al. (2019), called *follow the meta-leader*: $\mathbf{FTML}(\theta_t, \{\mathcal{T}_i\}_{i=1,\dots,t}) = \arg\min_\theta \left\{ \frac{1}{t}\sum_{k=1}^{t} \mathcal{L}(Alg(\mathcal{D}_k^{tr}, \theta), \mathcal{D}_k^{test}) \right\}$.

Teacher's decision-making task. The tutoring actions of the teacher correspond to the choice of a task to present to the learner: $\mathcal{A} = \{\mathcal{T}_i\}_{i=1,\dots,M}$, since they directly affect θ_0. Once a task $\mathcal{T}$ is chosen, the entire training dataset $\mathcal{D}^{tr}$ for $\mathcal{T}$ is used. Then the teacher has only the objective (O3) to consider. We choose to model the cost as the Euclidean distance to θ_0^* denoted by $d(\theta, \theta_0^*)$. It would be possible to include an inner-loop teacher that optimizes (O1) given $\mathcal{D}^{tr}$ further, but this makes it harder to demonstrate the benefit of optimizing (O3), thus we chose not to.

Algorithm. The interaction again defines a sequential leader-follower game. The teacher, as the leader, chooses which task to add to the current sequence of tasks. The learner responds by applying the FTML algorithm to update its initialization θ_0. The Stackelberg equilibrium for the stage game at time $t+1$ can be computed by solving the following bi-level optimization task:

$$\min_{\mathcal{T}} \; d(\theta, \theta_0^*) \quad \text{s.t.} \quad \theta \in \mathbf{FTML}(\theta_t, \{\mathcal{T}_i\}_{i=1,\dots,t} \cup \mathcal{T}).$$

FTML is a myopic follower and the dynamics are fully controlled by the leader's policy. Either of these properties sufficiently admits a dynamic programming solution to the computation of a strong Stackelberg equilibrium (Bucarey et al. 2019). Our rollout approximation uses one-step lookahead minimization and chooses the task that minimizes $d(\theta_{t+1}, \theta^*)$ at time t by applying the difficulty and usefulness decomposition given by Liu et al. (2017a) on the meta-gradient.

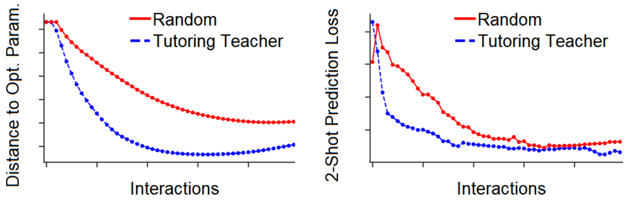

Figure 3: Left: by optimizing the choice and order of tasks, we can guide the online meta-learner towards a good initialization. Right: TtL leads to faster improvements on two-shot prediction loss for the online meta-learner.

Experimental Results

Setup. We generated 100 randomly selected non-linear regression tasks by using the class of sine functions as described in (Finn, Abbeel, and Levine 2017). The meta-learner employs a neural network and we aim to find a good initialization $\theta_0 \in \Theta$ for this network. Here, Θ is a real-valued vector space and $d(.,.)$ is the Euclidean distance. We first trained a neural network to perform regression using all 100 tasks with model-agnostic meta-learning and took the resulting initialization of this offline-trained neural network as θ^*, the optimal network initialization we would like to guide a learner towards. The learner employs the online meta-learning method with the follow-the-meta-leader algorithm (Finn et al. 2019). We have limited the number of tasks to 50, where the online meta-learner receives 50 tasks from the set of 100 training tasks sequentially. All experiments are conducted with 10 seeds and mean results are reported for visual clarity. Standard deviations are provided in the supplement.

Result. Figure 3 shows that TtL is able to guide the online meta-learner towards θ^*, which leads to quick improvements in 2-shot prediction loss compared to random task selection. The 2-shot prediction loss is evaluated by a test task the network has never seen before, randomly sampled from the distribution over sine functions.

Related Works

Machine teaching (Zhu 2015; Goldman and Kearns 1995) addresses the inverse problem of machine learning, where a teacher must select an optimal dataset to present to a learner. A machine teaching method aims to select a minimal dataset D such that the model $\theta = Alg(D)$ learned by a machine learner, based on algorithm Alg, is close to an optimal model θ^* (Zhu et al. 2018). An iterative variant (Liu et al. 2017a) assesses the iterative nature of some learning algorithms and shifts the problem from minimizing the size of a dataset to minimizing the number of steps. This method still assumes that the learner is fully-observed by the teacher (in particular that the learning algorithm is known) and that the teacher can only exchange data points. The method introduced by Liu et al. (2017b) alleviates these two problems, by considering that the learner and the teacher have different views of the same data and that the teacher does not know the algorithm of the learner, in a same way as proposed for the batch-version by Dasgupta et al. (2019). This is still different from what we propose, since they consider an unobserved but fixed algorithm for the learning, while our setting is built upon the possibility for the teacher to cause changes in the algorithm of the learner. Also, we do not restrict the actions of the teacher to the choice of data points. While Liu et al. (2017b) apply gradient-based methods, other alternatives have been proposed, based for instance on optimal control (Lessard, Zhang, and Zhu 2019), or models for sequential tasks where the learner is an inverse reinforcement learner (Cakmak and Lopes 2012; Haug, Tschiatschek, and Singla 2018; Parameswaran et al. 2019; Tschiatschek et al. 2019). A multi-agent formulation has been proposed by Hadfield-Menell et al. (2016) for teaching inverse reinforcement learners. In all these methods, the learner adapts to the teacher by updating only their estimated model and this line of work considers only the states of the world, whereas in our work we take one step further to considering the teacher's influence on the inner states of the learner (e.g. its priors, learning rate...) which affects *both* the learner's model and their learning algorithm. Finally, Peltola et al. (2019) proposed manipulative teaching of active sequential learners, where a manipulative teacher can steer the learner towards the parameters of its liking and showed that manipulation is more effective if the teacher has a model of the learner. However, this teaching strategy cannot achieve generalization on future tasks.

Multiple human teaching tasks have been formulated in terms of MDPs or POMDPs. In particular, the method proposed by Fan et al. (2018) considers that the teacher uses an MDP to adapt its teaching policy to the learner during the teaching process. In the domain of Intelligent Tutoring Systems, the use of multi-arm bandits has been suggested by Clément et al. (2015) as a way to adapt to multiple types of learners. As an alternative, POMDPs have been proposed to alleviate the uncertainty over the learner's cognitive state (Rafferty et al. 2016). Unlike our method, these papers only consider adapting to various profiles of learners, but do not consider the possibility of switching from one to another.

Conclusion

We proposed a novel mathematical framework that generalizes machine teaching to learners who can change their inductive biases. This induces a novel multi-objective approach to teaching, which can model problems that involve both machines and humans. This setting opens up various future research directions. From a theoretical point of view, it extends the question of teaching dimension to the minimal number of interactions necessary to teach without data manipulation to reach the optimal model. For practical applications, our method presents a rigorous mathematical framework that can be used to design interactive pedagogical systems for humans. Designers can choose to define the learner's internal state space and transition dynamics, or these can be learned interactively as well. The learner's internal state dynamics can also be learned with Bayesian reinforcement learning, where the teacher's decision-making task becomes a Bayes-adaptive POMDP instead.

Broader Impact

The proposed contribution can be seen from two different perspectives: teaching of machines and teaching of humans. Teaching of machines is intrinsically related to meta-learning and to the possibility of making a machine learner able to choose its algorithm by itself.

In the context of teaching human learners, which is on the rise with the emergence of Intelligent Tutoring Systems (ITS) (du Boulay 2016), the question of designing high-quality artificial teachers is a priority. However, as exposed in (Cochran-Smith 2003), even if there is a consensus on the need for good-quality teachers, the characteristics of good teaching are less clear. In a public opinion poll (Hart and Teeter 2002), it has been observed that only 19% of the participants mentioned that good-quality teaching entailed for the teacher to have a thorough understanding of the subject, against 42% for designing learning activities that inspired pupil interest. This observation highlights the perceived importance of pedagogy and points out that a teacher with only excellent knowledge would not be sufficient. The proposed framework alleviates this question, based on three considerations: (1) The thorough understanding of the subject is modelled by the access to θ^*, but teaching θ^* to the learner is not the sole priority unlike in standard machine teaching for instance; (2) The teacher plans a sequence of interactions with the learner, which corresponds to an understanding of teaching in the long-term; (3) The priority of the teacher is to help the learner progressing in their understanding. Thus, the framework we propose paves the way for high-quality automatic teaching. An important consideration is the conception of the models of learners, which needs to be learned automatically from observed interactions, or designed by human experts. An inaccurate choice for the model family can have harmful consequences, since seemingly innocent advice may lead to unexpected behaviours. As an illustration, the study proposed in (McNee, Kapoor, and Konstan 2006) shows that one irrelevant recommendation is enough to lose the trust of the user. Such a phenomenon would be of dramatic importance in a context of teaching.

Teaching human learners need not be limited to interactive tutoring systems though. The illustrating example and Case I illustrate the possibility of advanced modelling tools for scientists who are not expert statisticians, but use statistical analysis to draw conclusions from data. Such assistants could help scientists design statistical models by identifying the need for technical explanations and by sorting the relevant information from the data. In these domains, guaranteeing a non-manipulative teaching is of major importance, so that the users can gain and maintain a perfect understanding of their data. As such, the problem is very close to the question of understandability of Automatic ML (AutoML). Recent studies show that interpretability and visualization are key elements requested by users of AutoML systems (Drozdal et al. 2020). Our method would increase the understandability of such systems by making the users participate in the choice of the model and providing them explanations on modelling.

Finally, even though our work takes an important step towards the direction of non-manipulative teaching, we still need further research to protect learners against manipulative teaching algorithms. Even if we can guarantee to detect a naive manipulative teacher who would impose a model by force by selecting data, we have no guarantee over a teacher who would adapt their target model θ^* to pretend to be non-manipulative.

Acknowledgments

This work was supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence, FCAI; grants 319264, 313195, 305780, 292334, 328400, 28400), the UKRI Turing AI World-Leading Researcher Fellowship EP/W002973/1, KAUTE Foundation, and the Aalto Science-IT Project. Mustafa Mert Çelikok is partially funded by a personal grant from the Finnish Science Foundation for Technology and Economics (KAUTE). We acknowledge the computational resources provided by the Aalto Science-IT Project

References

Bertsekas, D. 2019. *Reinforcement learning and optimal control*, 69–80. Belmont, Massachusetts: Athena Scientific. ISBN 978-1886529397.

Bucarey, V.; Della Vecchia, E.; Jean-Marie, A.; and Ordóñez, F. 2019. Stationary Strong Stackelberg Equilibrium in Discounted Stochastic Games. Research Report RR-9271, INRIA.

Cakmak, M.; and Lopes, M. 2012. Algorithmic and human teaching of sequential decision tasks. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Chen, Y.; Singla, A.; Mac Aodha, O.; Perona, P.; and Yue, Y. 2018. Understanding the role of adaptivity in machine teaching: The case of version space learners. In *Advances in Neural Information Processing Systems*, 1476–1486.

Clément, B.; Roy, D.; Oudeyer, P.-Y.; and Lopes, M. 2015. Multi-Armed Bandits for Intelligent Tutoring Systems. *Journal of Educational Data Mining*, 7(2): 20–48.

Cochran-Smith, M. 2003. Teaching Quality Matters. *Journal of Teacher Education*, 54(2): 95–98.

Corbett, A. T.; and Anderson, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4): 253–278.

Dasgupta, S.; Hsu, D.; Poulis, S.; and Zhu, X. 2019. Teaching a black-box learner. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 1547–1555. Long Beach, California, USA: PMLR.

De Mol, C.; Giannone, D.; and Reichlin, L. 2008. Forecasting using a large number of predictors: Is Bayesian shrinkage a valid alternative to principal components? *Journal of Econometrics*, 146(2): 318–328.

Drozdal, J.; Weisz, J.; Wang, D.; Dass, G.; Yao, B.; Zhao, C.; Muller, M.; Ju, L.; and Su, H. 2020. Trust in AutoML: exploring information needs for establishing trust in automated machine learning systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*, 297–307.

du Boulay, B. 2016. Artificial intelligence as an effective classroom assistant. *IEEE Intelligent Systems*, 31(6): 76–81.

Fan, Y.; Tian, F.; Qin, T.; Li, X.; and Liu, T. 2018. Learning to Teach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1126–1135. JMLR. org.

Finn, C.; Rajeswaran, A.; Kakade, S.; and Levine, S. 2019. Online Meta-Learning. In *International Conference on Machine Learning*, 1920–1930.

Ghosh, J.; and Ghattas, A. E. 2015. Bayesian variable selection under collinearity. *The American Statistician*, 69(3): 165–173.

Goldman, S.; and Kearns, M. 1995. On the Complexity of Teaching. *Journal of Computer and System Sciences*, 50(1): 20 – 31.

Griffiths, T. L.; Callaway, F.; Chang, M. B.; Grant, E.; Krueger, P. M.; and Lieder, F. 2019. Doing more with less: meta-reasoning and meta-learning in humans and machines. *Current Opinion in Behavioral Sciences*, 29: 24–30.

Hadfield-Menell, D.; Russell, S. J.; Abbeel, P.; and Dragan, A. 2016. Cooperative inverse reinforcement learning. In *Advances in neural information processing systems*, 3909–3917.

Hamilton, J. D. 1989. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the Econometric Society*, 357–384.

Hart, P. D.; and Teeter, R. M. 2002. *A national priority: Americans speak on teacher quality*. Educational Testing Service.

Haug, L.; Tschiatschek, S.; and Singla, A. 2018. Teaching inverse reinforcement learners via features and demonstrations. In *Advances in Neural Information Processing Systems*, 8464–8473.

Lessard, L.; Zhang, X.; and Zhu, X. 2019. An Optimal Control Approach to Sequential Machine Teaching. In Chaudhuri, K.; and Sugiyama, M., eds., *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, 2495–2503. PMLR.

Liu, W.; Dai, B.; Humayun, A.; Tay, C.; Yu, C.; Smith, L. B.; Rehg, J. M.; and Song, L. 2017a. Iterative machine teaching. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2149–2158. JMLR. org.

Liu, W.; Dai, B.; Li, X.; Liu, Z.; Rehg, J. M.; and Song, L. 2017b. Towards black-box iterative machine teaching. *arXiv preprint arXiv:1710.07742*.

McNee, S. M.; Kapoor, N.; and Konstan, J. A. 2006. Don't look stupid: avoiding pitfalls when recommending research papers. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, 171–180.

Mei, S.; and Zhu, X. 2015. Using machine teaching to identify optimal training-set attacks on machine learners. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Parameswaran, K.; Devidze, R.; Cevher, V.; and Singla, A. 2019. Interactive Teaching Algorithms for Inverse Reinforcement Learning. In *The 28th International Joint Conference on Artificial Intelligence, 2019.*, CONF.

Patil, K. R.; Zhu, J.; Kopeć, Ł.; and Love, B. C. 2014. Optimal teaching for limited-capacity human learners. In *Advances in neural information processing systems*, 2465–2473.

Peltola, T.; Çelikok, M. M.; Daee, P.; and Kaski, S. 2019. Machine Teaching of Active Sequential Learners. In *Advances in Neural Information Processing Systems*, 11202–11213.

Rafferty, A. N.; Brunskill, E.; Griffiths, T. L.; and Shafto, P. 2016. Faster teaching via pomdp planning. *Cognitive science*, 40(6): 1290–1332.

Shafto, P.; Goodman, N. D.; and Griffiths, T. L. 2014. A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive psychology*, 71: 55–89.

Thrun, S.; and Pratt, L. 2012. *Learning to learn*. Springer Science & Business Media.

Tschiatschek, S.; Ghosh, A.; Haug, L.; Devidze, R.; and Singla, A. 2019. Learner-aware teaching: Inverse reinforcement learning with preferences and constraints. In *Advances in Neural Information Processing Systems*, 4147–4157.

Vanschoren, J. 2019. Meta-learning. In *Automated Machine Learning*, 35–61. Springer.

Zhu, X. 2015. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Zhu, X.; Singla, A.; Zilles, S.; and Rafferty, A. N. 2018. An overview of machine teaching. *arXiv preprint arXiv:1801.05927*.