

Machine Coaching with Proxy Coaches

Vassilis Markos¹, Marios Thoma¹ and Loizos Michael^{1,2}

¹Open University of Cyprus, Nicosia, Cyprus

²CYENS Centre of Excellence, Nicosia, Cyprus

Abstract

We evaluate the *Machine Coaching* paradigm, a human-in-the-loop machine learning methodology, according to which a human coach and a machine engage in an iterative bidirectional exchange of explanations, towards improving the machine's ability to reach conclusions and justify them in a way that is acceptable to the human coach. To support the systematic empirical investigation of the efficacy and efficiency of Machine Coaching, we adopt proxy (algorithmic) coaches in the stead of human ones.

Keywords

machine coaching, proxy coaching, explainable AI

1. Introduction

Learning in humans proceeds in many and diverse ways. Under what could be called the *autodidactic paradigm* [1, 2], the human learner utilizes whatever information is available. A human supervisor, if present, may complete / enrich that information, so that the human learner faces a more benign / informative environment for learning.

Another, significant, part of human learning takes place under what could be called the *coaching-based paradigm* [3, 4], where a human supervisor, or coach, shares with the human learner not only *what* the case is in a certain state of the environment, but chiefly *why* that case is. This happens whenever parents warn their children to “not run while holding scissors, because they will hurt themselves”, whenever teachers explain to their students to “conclude that two triangles are similar because they have congruent angles”, when chess instructors teach their pupils to “place pieces on squares from which they cannot be easily deflected”, and when managers direct their assistants to “book a hotel close to the meeting venue for same-day trips”.


Such *pieces of advice* from the coach do not typically come unprompted, but as a reaction to a wrong or wrongly-justified decision by the learner. A coach offering advice is, in effect, proactively completing / enriching missing information in states of the environment that the learner might encounter, by providing conditions on states under which a certain decision should be reached. Compared to the autodidactic paradigm, the substantial amount of information communicated under the coaching-based paradigm is conducive to more efficient, albeit coach-specific, learning, while the reactive nature of advising entails only marginally extra effort

ArgML 2022: 1st International Workshop on Argumentation and Machine Learning @COMMA 2022, September 13, 2022, Cardiff, Wales, UK

✉ vasileios.markos@st.ouc.ac.cy (V. Markos); marios.thoma@st.ouc.ac.cy (M. Thoma); loizos@ouc.ac.cy (L. Michael)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

from the coach, as humans are known to be competent at identifying counter-arguments when challenged [5].

*Machine Coaching*¹ [3, 4] was proposed as an argumentation-based learning-theoretic framework under which the coaching-based paradigm of learning can be studied. It formalizes the computational resources available to the coach and the learner, the protocol and language of their interaction, and the expected quality guarantees on what is learned. Under that framework, one considers a human coach, with access to a target policy, interacting with a machine learner, and formally establishes that if the human coach offers appropriate, in a defined-sense, advice, then the machine learner can utilize that advice to learn a hypothesis policy that approximates the target policy with the expected quality guarantees.

Undertaking a study with human participants in the role of the coach is, clearly, the ultimate way to empirically validate the Machine Coaching framework. Yet there are three major challenges to overcome in such an envisioned study.

The first challenge stems from *the lack of fluency of average humans in formal logic*, making it hard for them to exchange explanations in the native (logic-based) language of Machine Coaching. This issue could be alleviated by adopting a natural language interface for exchanging explanations, which would be automatically translated to/from the native language. Although there is some work in this direction — even using the Machine Coaching framework itself at a meta-level [6] — the problem is sufficiently complex, and currently lacks a robust solution.

The second challenge stems from *the inaccessibility of the target policy that humans have in mind when coaching*. Without a target policy, one cannot empirically evaluate the quality of the learned hypothesis policy. If, in addition, no policy agrees with all the advice offered by the human coach, then the empirical study would confound the validation of the Machine Coaching framework with the determination of the expressivity of the framework’s native language, and with the compatibility of the interaction with human cognition.

The third challenge stems from *the physical and mental limitations of humans in prolonged interactions*. A systematic evaluation of the Machine Coaching framework would require substantial rounds of interactions between the coach and the machine, across diverse target policies. It is unlikely that humans would maintain efficient and consistent behavior over time, either due to fatigue, an unconscious adaptation to the empirical setting, or a suppression of their natural reactions when coaching with an externally-given target policy.

To eschew these challenges, we resort to the use of a *proxy (algorithmic) coach* that can communicate in the native language of Machine Coaching, cope with a given explicit target policy, and provide advice efficiently and consistently. These features need to be balanced against the desire to stay close to human coaches, who have only approximate access to their target policy — conceivably as a compilation, in some intensional or extensional form, of their relevant life experiences — and are unable to divulge the target policy on cue, but can still react if some part of it is “challenged” by an external position and associated argument [5]. Though the proxy coaches introduced below do not claim to serve as a comprehensive simulation of human ones, they aim to capture the interaction aspects that are important for a sound *in vitro* assessment of Machine Coaching. Consequently, we consider our work as a stepping stone between the theoretically proven efficiency of Machine Coaching and an empirical assessment

¹Resources available at: <https://cognition.ouc.ac.cy/prudens>.

of that through a study involving actual human participants.

Accordingly, we consider proxy coaches with access to an *exemplar set* of data, labeled with the predictions of a fixed (but hidden to the proxy coaches) target policy. *The key technical question, then, is how to extract appropriate pieces of advice from the exemplar set.* In the sequel we seek to answer this question by demonstrating how to develop proxy coaches with both an intensional and an extensional compilation of the exemplar data, and continue to evaluate them empirically. We shall remark at this point that our focus in this paper is mostly shifted towards investigating the various alternatives of proxy coaches and their *qualitative* differences. Consequently, while we do present and discuss *quantitative* results for all chosen proxy coaches, they mostly serve as a means to stress the effects each proxy coaching protocol has on the coaching process.

2. Background and Related Work

The process whereby an agent learns under the supervision of a more experienced tutor / coach, who offers advice as a reaction to the learner’s decisions or actions, appears as a suggestion for the development of AI systems in John McCarthy’s seminal work on the “Advice Taker” [7]. Recent work [3, 4] has sought to formalize this process, termed *Machine Coaching*, in learning-theoretic terms, as a variant of the Probably Approximately Correct (PAC) model [8].

The eXplainable Interactive Learning (XIL) framework [9] also considers a human supervisor in the role of a learning “coach”, providing advice either in the form of a (more) correct label, or a correct explanation. Unlike Machine Coaching, which adopts learning and reasoning semantics compatible with the language of formal argumentation, XIL assumes black-box access to an active learning algorithm [10] and a local post-hoc explainer [11]. Correspondingly, the explanations provided by the “coach” in XIL cannot be directly and elaboration-tolerantly [12] embedded into the learned model, but are rather used to create additional labeled data for the further training of an opaque learned model.

The idea of online ingestion of human advice in Machine Learning processes is present in other works as well. A way to incorporate user advice into Support Vector Machines is proposed in [13], using the advice to reduce the number of data points that need to be labeled. In [14], an interactive framework is proposed that allows human users to label selected data points for the purpose of training a document classifier. Similarly, in [15], a learning paradigm is proposed that systematically utilizes user-provided explanations to reduce learning complexity, stressing that data richness (i.e., including explanations along with labels) over data volume (i.e., having access to many labeled data without explanations) leads, in certain cases, to equally good or better performance.

Beyond speeding up learning, more recent works examine how human knowledge may benefit the entire learning process, by being an integral part of it. *Coactive Learning*, proposed in [16], is an interactive Machine Learning paradigm where a learner receives sub-optimal user advice that is, inherently, only “slightly better” than the learner’s current prediction. However, it is shown that several existing supervised algorithms can be altered to accommodate this type of human-machine interaction.

Although our work herein focuses on the empirical evaluation of Machine Coaching, the

technique of proxy coaches that we follow would also seem to be applicable, to varying extents, to the frameworks above. The rest of the works that we briefly review below are not tied to coaching per se, but relate to our chosen implementation of the proxy coaches.

In [17], the authors attempt to offer post-hoc explanations for a black-box model by training a random forest, extracting rules from it, and using them to construct an argumentation theory. One of our considered proxy coach types also adopts the view that a random forest can be the source of arguments, but in our case these arguments are not the end product itself, but are rather the pieces of advice that are given from the proxy coach to the ultimate learner.

On the other hand, we also consider a proxy coach type that does not compile the available data into an explicit learned model, but uses them only implicitly. This approach relates to the works in [18, 19], which investigate a paradigm of implicit learning, where one can reason and respond to queries by directly consulting the data. Whereas the emphasis of those works is on answering given queries, our emphasis is on constructing appropriate pieces of advice for the proxy coach to offer to the ultimate learner.

To facilitate the choice of appropriate advice in this implicit learning setting, we use natural selection over an evolutionary process. The formalization of evolution that we adopt is that in [20], where evolution is cast as a learning problem that seeks to approximate a hidden target function. The evolutionary process in our case attempts to identify the next appropriate piece of advice, and to integrate it into the next generation of organisms, with each organism encoding a collection of rules aimed to approximate the entire target policy. Thus, our approach resembles a Pittsburgh-style Learning Classifier System [21].

3. From Direct to Proxy Coaches

Following the Machine Coaching framework, a (direct) coach and a learner hold, respectively, a fixed *target policy* and a revisable *hypothesis policy*, each represented in the form of a partially-ordered set of “if-then” rules. Upon perceiving a new *context*, the learner returns the predictions of the current hypothesis policy, and an *explanation* in the form of hypothesis policy rules that support those predictions. The coach responds by offering *advice* in the form of target policy rules that support *why* the learner’s predictions were incorrect, incomplete, or improperly-justified according to the coach. The learner revises the hypothesis policy, and the process repeats.

While arguments within Machine Coaching are generally considered to be arbitrary trees of rules, for the sequel, we restrict our attention to *shallow* propositional policies, whose rules have a special atom (or its negation) as their head, so that rules are not chained during reasoning — and, thus, each argument comprises of a single rule. Relatedly, we restrict our attention to *complete* contexts, which specify truth-values for all remaining atoms. Without loss of generality, we hence let the special atom be output.

Thus, if output were the ability to fly, and given a context {penguin, bird}, the learner could predict output by offering the following explanation: “bird implies output”, and the coach could react by offering the following piece of advice: “penguin implies -output”, which would be integrated with higher priority in the revised hypothesis policy.

Even with these restrictions, policies can still vary along other dimensions, which affect the

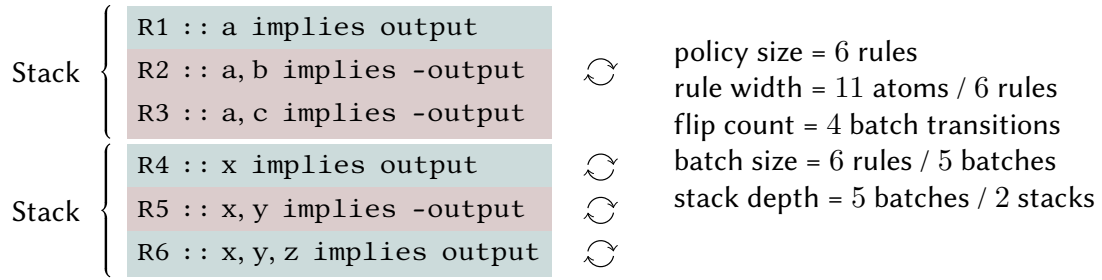


Figure 1: A shallow propositional policy over a set of atoms $A \supseteq \{a, b, c, x, y, z\}$. Rules increase in priority from top to bottom. Colored regions indicate batches; circular arrows indicate flips.

conceptual and computational complexity of learning (through Machine Coaching, or otherwise) and reasoning. Policies can vary in terms of their **policy size**, which counts the number of rules in the policy, their **rule width**, which counts the average number of atoms in rule bodies, their **flip count**, which counts the flips / transitions between **batches** of consecutive rules with the same head polarity (positive or negative), the **batch size**, which counts the average number of rules across batches, and the **stack depth**, which counts the average number of consecutive batches such that rules in a batch have logically more specific conditions than the rules in the preceding batch (cf. Figure 1).

Proxy coaches have only indirect knowledge of some target policy p , by being given access to an **exemplar set** \hat{p} of contexts labeled according to p . In broad terms, then, a proxy coach faces a context $x \in \mathcal{C}$ from some **coaching set** \mathcal{C} , receives the prediction / explanation $h(x)$ of the learner’s current hypothesis policy h on x , and then uses \hat{p} to generate a piece of advice $\alpha(x, h(x); \hat{p})$ to be offered to the learner. An **epoch** concludes whenever a piece of advice is offered, noting that the proxy coach need not offer advice for each $x \in \mathcal{C}$.

The precise manner on how $\alpha(x, h(x); \hat{p})$ is generated depends on the proxy coach type, but it generally seeks to consider the anticipated effect that each candidate piece of advice would have on the learner’s hypothesis policy, in terms of: (i) improving its **coverage**, by reducing the number of contexts on which it abstains; and (ii) improving its **accuracy**, by reducing the ratio of wrong over correct predictions it makes.

3.1. Intensional Proxy Coaches

The first class of proxy coaches that we consider encode intensionally their knowledge of the target policy, by using the exemplar set \hat{p} once, before the coaching phase, to supervise the training of a white-box model m . The ante-hoc explainability of m makes it, then, natural to seek to “extract” a piece of advice $\alpha(x, h(x); \hat{p})$ from the explanation of $m(x)$, given a context $x \in \mathcal{C}$. *The exemplar set, therefore, takes the role of a training set.* Coaching proceeds, then, as in Algorithm 1.

Our chosen white-box models are those of decision trees and random forests, from which we have identified three natural ways of extracting advice as part of Step 6 of Algorithm 1.

Our first approach considers a white-box model m based on a single decision tree. Given the current context $x \in \mathcal{C}$, it identifies the active path of the tree, from which it computes, as usual,

Algorithm 1 Intensional Proxy Coaching

- 1: **input:** exemplar set \hat{p} ; coaching set \mathcal{C} .
 - 2: Train white-box model m , using \hat{p} as training instances.
 - 3: **for** the next context $x \in \mathcal{C}$ **do**
 - 4: Ask for learner’s prediction / explanation $h(x)$.
 - 5: **if** predictions of $m(x)$ and $h(x)$ differ **then**
 - 6: Get advice from explanations of $m(x)$ and $h(x)$.
 - 7: Give advice to revise current hypothesis policy h .
 - 8: **end if**
 - 9: **end for**
-

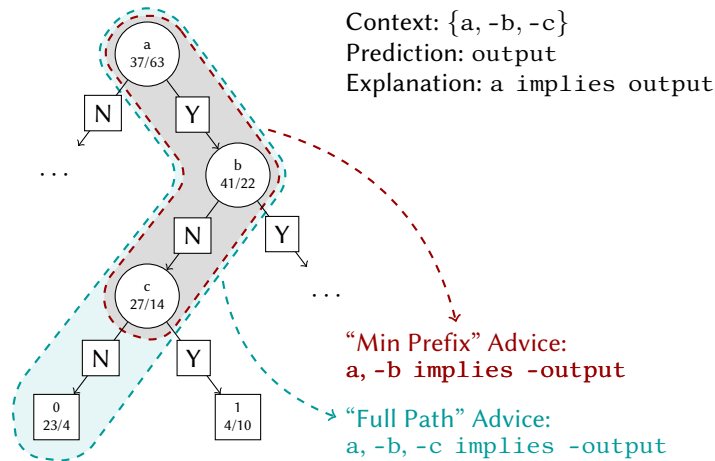


Figure 2: Examples of two ways of extracting advice from a path in a decision tree. Internal nodes indicate the atom based on which a split is made, and leaf nodes indicate the prediction of the corresponding path. The two numbers at the bottom of a node correspond to the negatively / positively labeled training instances that reach that node.

the prediction of $m(x)$. The full path itself acts as the explanation of $m(x)$, and is returned as a piece of advice. By construction, the advice so generated is “cautious”, in that a full path in decision trees tends to have high accuracy on the exemplar set, as otherwise the decision tree learning algorithm would have chosen to expand the path further. For the same reason, however, the advice is also “specific”, in that it applies only on a few contexts because of the path length. Overall, then, this type of a proxy coach chooses to provide advice based on the expectation that the hypothesis policy will improve its coverage only marginally, but it will be highly effective in improving and maintaining its accuracy.

Our second approach is a variant of the first, where instead of considering the full active path of the decision tree as the (unique candidate for a) piece of advice, it considers all of its prefixes as candidates. Among them, it chooses the minimal one that is simultaneously more specific than the explanation of $h(x)$, and such that the prediction of the decision tree would have remained the same had the active path been pruned to be that particular prefix. By construction, the advice so generated is more “loose”, in that a prefix of a path in decision

Algorithm 2 Extensional Proxy Coaching

- 1: **input:** exemplar set \hat{p} ; coaching set \mathcal{C} .
 - 2: Initialize the current hypothesis policy h to be empty.
 - 3: **for** the next context $x \in \mathcal{C}$ **do**
 - 4: Create mutations of types $(M0)$, $(M+)$, $(M\downarrow)$.
 - 5: Compute fitness score f_j of each mutation h_j .
 - 6: Select “good” beneficial or neutral mutation h_i .
 - 7: Set the current hypothesis policy to be h_i .
 - 8: **end for**
-

trees tends to have lower accuracy on the exemplar set, which is why the decision tree learning algorithm had chosen to expand the prefix further. For the same reason, however, the advice is also more “general”, in that it applies on more contexts because of the shorter prefix length. Overall, then, this type of a proxy coach chooses to provide advice based on the expectation that the hypothesis policy will improve its coverage measurably, but at the cost of being less effective in improving and maintaining its accuracy.

Our third approach considers a white-box model m based on a random forest. Given the current context $x \in \mathcal{C}$, it computes, as usual, the prediction of $m(x)$. It then proceeds to identify the active path from each individual tree whose prediction matches that of $m(x)$, and considers those full paths as candidates. There are many strategies for choosing one of the candidates, but for concreteness, our particular approach chooses the one that is most “specific” (breaking ties randomly), in that it is activated by as few contexts as possible in the exemplar set. Given that paths in trees in random forests are typically short and hence “general”, the approach tends to favor the generation of a “maximally cautious” advice among “typically loose” candidates. Overall, then, this type of a proxy coach chooses to provide advice based on the expectation of striking a balance between the improvement of the coverage and accuracy of the hypothesis policy.

3.2. Extensional Proxy Coaches

The second class of proxy coaches that we consider retain extensionally their knowledge of the target policy, without compiling it into another form. Rather, they use the exemplar set \hat{p} repeatedly, during the coaching phase, to decide whether a candidate piece of advice should become $\alpha(x, h(x); \hat{p})$, given a context $x \in \mathcal{C}$. *The exemplar set, therefore, takes the role of a validation set.* Coaching proceeds, then, as in Algorithm 2.

This extensional perspective suggests a trial-and-error view of coaching, where the coach tests candidate pieces of advice to measure their effect. Since a systematic testing of all possible pieces of advice is infeasible, the challenge for the coach is to select the set of candidates, and to identify how to test the effect of each candidate on the hypothesis policy.

Our approach adopts an evolutionary mechanism, where, roughly, the process of mutation corresponds to the generation of the candidate pieces of advice, and the process of natural selection corresponds to their testing towards selecting $\alpha(x, h(x); \hat{p})$. We discuss below in more detail the nuances that come from the fact that the same evolutionary mechanism needs to

simulate both the proxy coach and the learner.

At the start of a generation, the population comprises the current hypothesis policy h . Given the current context $x \in \mathcal{C}$, certain candidate pieces of advice are considered, and each is “provisionally” offered as advice to a different copy of h , giving rise to its offspring. Specifically: ($M0$) an offspring is created by offering no advice; ($M+$) an offspring is created by offering the advice “ x implies output”, if $h(x)$ predicts -output or abstains; ($M+$) an offspring is created by offering the advice “ x implies -output”, if $h(x)$ predicts output or abstains; ($M\downarrow$) an offspring is created, for each j , by offering the advice “body $_{-j}$ implies head”, where “body implies head” is the latest advice that was integrated in h , and body $_{-j}$ is body minus its j -th literal.

These aforementioned pieces of advice are said to be offered “provisionally” in the sense that the proxy coach may, at a subsequent generation, seek to refine a previously given piece of advice by means of type ($M\downarrow$) mutations. A piece of advice is “conclusively” given at the end of a streak of (zero or more) type ($M0$) or ($M\downarrow$) mutations that follow a type ($M+$) mutation. This is the point at which an epoch concludes.

At each generation, exactly one of the offspring is chosen to survive to initialize the next generation. To support this selection process, each offspring h_i is evaluated in terms of its improvement in accuracy and coverage relative to its parent against each exemplar $e \in \hat{p}$. The change from $h(e)$ to $h_i(e)$ is considered: positive, if a wrong prediction changes to a correct one or an abstention, or an abstention changes to a correct prediction; negative, if a correct prediction changes to a wrong one or an abstention, or an abstention changes to a wrong prediction. By giving a +1 or -1 fitness point to offspring h_i for each respective positive or negative change across the exemplar set \hat{p} , we end up with an intuitive metric f_i that aggregates the improvement effect of the advice given to offspring h_i on both its coverage and its accuracy.

The offspring are, then, grouped based on the relation of their relative fitness to a fixed threshold parameter t . An offspring h_i is **detrimental**, **neutral**, **beneficial** if its relative fitness f_i belongs in $(-\infty, -t)$, in $[-t, +t]$, in $(+t, +\infty)$, respectively. Among the beneficial ones, if available, the offspring h_i is selected to survive with probability $f_i^k / \sum_j f_j^k$, where the exponent k is a fixed non-linearity parameter. Otherwise, a neutral offspring (whose existence is guaranteed by a type ($M0$) mutation) is selected uniformly at random.

The proposed approach searches greedily for advice that is as “cautious” and as “general” as possible, while starting the greedy search from a fully “cautious” and “specific” seed advice selected by a type ($M+$) mutation. Overall, then, this type of a proxy coach chooses to provide advice based on the expectation of striking a balance between the improvement of the coverage and of the accuracy of the hypothesis policy.

4. Empirical Investigation

We present below the key aspects and results of our empirical investigation. Additional details are given in the appendices. All related materials may be found at <https://github.com/VMarkos/proxy-coaching-argml-2022>.

4.1. Empirical Setting

We first fixed a set A of 20 atoms (other than output) for use in contexts and rule bodies. We then constructed 10 groups with 20 policies each, with each pair of groups corresponding to high versus low values in one of the variability dimensions discussed in Section 3, giving rise to a set P of 132 distinct target policies. For each target policy $p \in P$, we constructed an associated exemplar set \hat{p} by uniformly at random sampling 1000 contexts from 2^A , labeling each context according to p , filtering out any contexts on which p abstained, and keeping 70% of the labeled contexts. The other 30% was used to populate an evaluation set \mathcal{E} . An additional 500 (unlabeled) contexts were sampled to populate a coaching set \mathcal{C} .

We ran an experiment for each pairing of a target policy $p \in P$ with a proxy coach type discussed in Section 3: (E1) a decision tree with “full path” advice, (E2) a decision tree with “min prefix” advice, (E3) a random forest, and (E4) an evolutionary mechanism. Decision trees in experiments (E1), (E2), and (E3) were trained on the exemplar set \hat{p} using ID3 [22]. Random forests in experiments (E3) comprised 20 trees, each trained on a 10% random fraction of \hat{p} . The evolutionary mechanism in experiment (E4) used a threshold parameter $t = 0$, and a non-linearity parameter $k = 2$.

The current hypothesis policy was examined at the end of each epoch in each experiment. *Performance* values recorded how many of the predictions of the current hypothesis policy on the evaluation set \mathcal{E} were correct against p , wrong against p , or abstentions, and the accuracy of the definite predictions (i.e., the number of correct predictions over the number of correct or wrong predictions). *Relative Size* values recorded the size of the current hypothesis policy (relative to the size of p), and the size of the epoch (relative to the size of \mathcal{C}), the size of each given piece of advice (relative to the size of A).

The above values for each proxy coach type were aggregated across P , and were plotted against the epochs. To account for different numbers of epochs across experiments, the set of epochs of each experiment was uniformly distributed over the interval $[0, 1]$, using spline-interpolation to fill-in the values between the points that corresponded to the epochs in that experiment. In particular, a value of 0 corresponds to the pre-coaching state of affairs, where the current hypothesis policy is empty, and a value of 1 corresponds to the post-coaching state of affairs, where the current hypothesis policy is the one after the integration of the final piece of advice.

4.2. Results and Analysis

Figure 3 presents plots for the four sets of experiments.

Machine Coaching Efficacy

A first general observation is that experiments agree qualitatively on the efficacy of Machine Coaching. As more pieces of advice are given, the performance of the current hypothesis policy improves in terms of increased correct predictions and decreased abstentions. The size of the current hypothesis policy grows linearly with the given pieces of advice, due to the uniform distribution of epochs on the X axis. The sizes of the epochs and the given pieces of advice

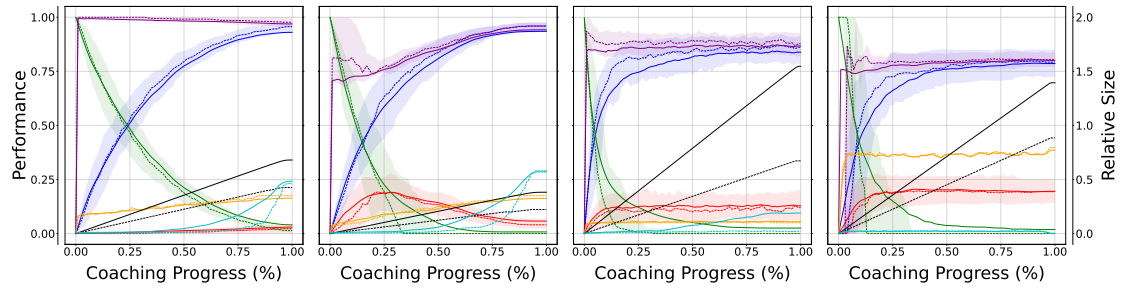


Figure 3: Results for experiments (E1)–(E4), shown from left to right. Performance values show correct predictions (—), wrong predictions (—), abstentions (—), and accuracy (—) on the evaluation set, against the target policy. Relative Size values show the relative sizes of the current hypothesis policy (—), the epoch (—), and the given piece of advice (—). Results are aggregated across all target policies, with solid lines representing the mean values, dashed lines representing the median values, and shaded areas representing the $Q1$ to $Q3$ quartile interval.

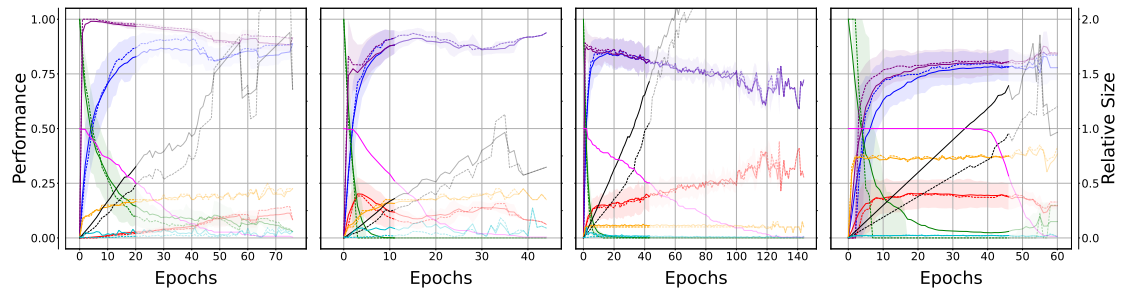


Figure 4: Results for experiments (E1)–(E4), as in Figure 3, with the exception that the X axis shows the epoch count instead of the coaching progress. The number of experiments for the various target policies that reach each epoch and participate in the analysis are also plotted (—). The lines are shown faded out once the number of participating experiments drops below 50% of the total number of experiments in each set.

remain steady or increase, confirming the intuition that advice is given less frequently and becomes more specific as coaching progresses.

Experiments (E1) and (E2)

The first set of experiments cleanly demonstrates the effect of offering “cautious” but “specific” advice. As expected, wrong predictions remain effectively zero throughout the coaching phase, while abstentions are very gradually replaced with correct predictions, maintaining an effectively perfect accuracy.

Analogously, the second set of experiments demonstrates the effect of offering more “loose” and “general” advice. Correct predictions increase slightly faster compared to experiments (E1), while abstentions decrease at an even faster pace, causing the introduction of wrong predictions, and a sub-perfect accuracy. Interestingly, most wrong predictions are corrected over time, leading to sufficiently high accuracy.

In both sets of experiments, advice becomes more specific over time and is given less fre-

quently, with the epoch size increasing significantly during the end of the coaching phase. These considerations, along with the continual improvement of performance, suggest that the given advice is indeed beneficial. The quality of the given advice is further supported by the size of the hypothesis policy remaining distinctly smaller than the size of the target policy, indicating that the given advice is able to compress parts of the target policy. The more “loose” and “general” advice in experiments (*E2*) seems to correspond to a less frequent need for advice, and a more concise hypothesis policy, at the expense of occasionally making wrong predictions, but also effectively never abstaining.

Experiments (*E3*) and (*E4*)

The third and fourth sets of experiments demonstrate the effect of offering advice chosen greedily through bounded local searching, and without consulting the explanations of the current hypothesis policy on its predictions. Accordingly, performance is measurably worse than in the first two sets of experiments, with fewer correct predictions, more wrong predictions, higher number of abstentions, and lower accuracy.

Although the last two sets of experiments demonstrate a more aggressive take on decreasing abstentions and increasing correct predictions earlier on, this seems to come at the expense of introducing a considerable number of wrong predictions that persist across epochs, indicating a lower quality of the given advice. This indication is further corroborated by the persistent size of the advice, which fails to become more “cautious” and “specific” over time, and likely leads to the introduction of nearly equally-many new wrong predictions in the place of any of the existing wrong predictions it corrects.

Relatedly, the epoch size does not increase as rapidly as in the first two sets of experiments, leading to a larger hypothesis policy size, to the extent, in fact, that the hypothesis policy ends up being larger than the target policy, showing an inability for compression, and induction, in the given advice.

Comparatively, the size of the given piece of advice in experiments (*E4*) is multiple times larger than that in experiments (*E3*). This can be directly attributed to the initialization of the search used in each case, with the former constructing advice by starting from a fully “cautious” and “specific” seed advice, and the latter selecting a piece of advice from typically “loose” and “general” candidate pieces of advice.

Machine Coaching Efficiency

The primary metric on efficiency for Machine Coaching is the number of epochs required to get to a certain degree of performance. Figure 4 shows the same information as Figure 3, but with aggregation happening over each particular epoch. Since different target policies lead to different numbers of resulting epochs, the aggregation happens over a diminishing set of target policies which eventually becomes unrepresentative of the initial set, and should not form the basis for conclusions.

Looking, therefore, at the parts of the plots that include at least 50% of the entire set of target policies being considered, we observe qualitatively that a few (around 10–20) pieces of advice seem to suffice to get relatively high performance.

Although reporting absolute computation times is not informative, we do remark that experiments (*E4*) are the most time-intensive ones, as a result of the repeated testing against the entire exemplar set, for each mutation in each generation.

5. Conclusions

We have put forward proxy (algorithmic) coaches as a means to provide empirical evidence that complements prior theoretical work on the efficacy and efficiency of Machine Coaching. In ongoing work, we continue to investigate coaching-based learning further with proxy coaches in more expressive settings (e.g., relational policies, richer reasoning semantics with rule chaining, and, relatedly, without assuming complete contexts), and contrast its performance and scalability against autodidactic learning algorithms that map exemplar data directly into a form of prioritized rules [23, 24, 25, 26].

Ultimately, our plan is to undertake empirical studies with humans in the role of direct coaches, by identifying meaningful solutions to the challenges discussed in Section 1, perhaps by pairing human coaches with proxy (algorithmic) coaches. Whether human coaches will offer advice in a manner analogous to one of the proxy coaches considered herein, whether they will remain consistent across multiple interactions, and whether they will find the coaching protocol to be cognitively light, are all questions that we wish to investigate. In turn, we expect that answers to these questions will offer guidelines towards improving the cognitive compatibility of the language, semantics, and protocol of Machine Coaching.

Acknowledgements This work was supported by funding from the European Regional Development Fund and the Government of the Republic of Cyprus through the Research and Innovation Foundation under grant agreement no. INTEGRATED/0918/0032, from the EU’s Horizon 2020 Research and Innovation Programme under grant agreements no. 739578 and no. 823783, and from the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation, and Digital Policy.

References

- [1] L. Michael, Autodidactic Learning and Reasoning, Ph.d. thesis, School of Engineering and Applied Sciences, Harvard University, USA, 2008. URL: <https://dl.acm.org/doi/abs/10.5555/1467943>.
- [2] L. Michael, Partial Observability and Learnability, *Artificial Intelligence* 174 (2010) 639–669. doi:10.1016/j.artint.2010.03.004.
- [3] L. Michael, The Advice Taker 2.0, in: *Proceedings of the 13th International Symposium on Commonsense Reasoning*, volume 2052, London, U.K, 2017. URL: <http://ceur-ws.org/Vol-2052/#paper13>.
- [4] L. Michael, Machine Coaching, in: *IJCAI 2019 Workshop on Explainable Artificial Intelligence*, Macau, China, 2019, pp. 80–86. URL: https://cognition.ouc.ac.cy/loizos/papers/Michael_2019_MachineCoaching.pdf.

- [5] H. Mercier, D. Sperber, Why Do Humans Reason? Arguments for an Argumentative Theory., *Behavioral and Brain Sciences* 34 (2011) 57–74. doi:10.1017/S0140525X10000968.
- [6] C. Ioannou, L. Michael, Knowledge-Based Translation of Natural Language into Symbolic Form, in: *Proceedings of the 7th Linguistic and Cognitive Approaches To Dialog Agents Workshop - LaCATODA 2021, Montreal, Canada, 2021*, pp. 24–32. URL: <http://ceur-ws.org/Vol-2935/#paper3>.
- [7] J. McCarthy, Programs with Common Sense, in: *Proceedings of the Teddington Conference on the Mechanization of Thought Processes, London, U.K, 1959*, pp. 75–91. URL: <http://jmc.stanford.edu/articles/mcc59/mcc59.pdf>.
- [8] L. G. Valiant, A Theory of the Learnable, *Communications of the ACM* 27 (1984) 1134–1142. doi:10.1145/1968.1972.
- [9] S. Teso, K. Kersting, Explanatory Interactive Machine Learning, in: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, AIES '19, Association for Computing Machinery, New York, NY, USA, 2019*, pp. 239–245. doi:10.1145/3306618.3314293.
- [10] B. Settles, Active Learning Literature Survey, Technical Report, University of Wisconsin-Madison Department of Computer Sciences, 2009. URL: <https://minds.wisconsin.edu/handle/1793/60660>.
- [11] M. T. Ribeiro, S. Singh, C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016*, pp. 1135–1144. doi:10.1145/2939672.2939778.
- [12] J. McCarthy, Elaboration Tolerance, in: *Common Sense 98, London, U.K, 1998*. URL: <http://www-formal.stanford.edu/jmc/elaboration.html>.
- [13] H. Raghavan, J. Allan, An Interactive Algorithm for Asking and Incorporating Feature Feedback into Support Vector Machines, in: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07, Association for Computing Machinery, New York, NY, USA, 2007*, pp. 79–86. doi:10.1145/1277741.1277758.
- [14] B. Settles, Closing the Loop: Fast, Interactive Semi-Supervised Annotation With Queries on Features and Instances, in: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK., 2011*, pp. 1467–1478. URL: <https://aclanthology.org/D11-1136>.
- [15] O. Zaidan, J. Eisner, C. Piatko, Using "Annotator Rationales" to Improve Machine Learning for Text Categorization, in: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, Association for Computational Linguistics, Rochester, New York, 2007*, pp. 260–267. URL: <https://aclanthology.org/N07-1033>.
- [16] P. Shivaswamy, T. Joachims, Coactive Learning, *Journal of Artificial Intelligence Research* 53 (2015) 1–40. doi:10.1613/jair.4539.
- [17] N. Prentzas, A. Nicolaides, E. Kyriacou, A. Kakas, C. Pattichis, Integrating Machine Learning with Symbolic Reasoning to Build an Explainable AI Model for Stroke Prediction, in: *2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), Athens, Greece, 2019*, pp. 817–821. doi:10.1109/BIBE.2019.00152.
- [18] R. Khardon, D. Roth, Learning to Reason, *Journal of the ACM* 44 (1997) 697–725. doi:10.

1145/265910.265918.

- [19] B. Juba, Implicit Learning of Common Sense for Reasoning, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, AAAI Press, Beijing, China, 2013, pp. 939–946. URL: <https://www.ijcai.org/Proceedings/13/Papers/144.pdf>.
- [20] L. G. Valiant, Evolvability, *Journal of the ACM* 56 (2009) 1–21. doi:10.1145/1462153.1462156.
- [21] R. J. Urbanowicz, J. H. Moore, Learning Classifier Systems: A Complete Introduction, Review, and Roadmap, *Journal of Artificial Evolution and Applications* 2009 (2009) 1–25. doi:10.1155/2009/736398.
- [22] J. R. Quinlan, Induction of Decision Trees, *Machine Learning* 1 (1986) 81–106. doi:10.1023/A:1022643204877.
- [23] R. L. Rivest, Learning Decision Lists, *Machine Learning* 2 (1987) 229–246. doi:10.1023/A:1022607331053.
- [24] Y. Dimopoulos, A. Kakas, Learning Non-Monotonic Logic Programs: Learning Exceptions, in: N. Lavrac, S. Wrobel (Eds.), *Machine Learning: ECML-95*, volume 912 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 1995, pp. 122–137. doi:10.1007/3-540-59286-5_53.
- [25] L. Michael, Causal Learnability, in: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, AAAI Press, Barcelona, Spain, 2011, pp. 1014–1020. doi:10.5591/978-1-57735-516-8/IJCAI11-174.
- [26] L. Michael, Cognitive Reasoning and Learning Mechanisms, in: Proceedings of the 4th International Workshop on Artificial Intelligence and Cognition, volume 1895 of *CEUR Workshop Proceedings*, CEUR, New York City, NY, 2016, pp. 2–23. URL: http://ceur-ws.org/Vol-1895/#AIC16_paper1.

A. Empirical Setting (Details)

We synthetically generated a target policy by starting from a set of atoms A , an average batch size b , a flip count ℓ , and a number of stacks s , and then partitioning $\ell + 1$ into s positive integers $d_1, \dots, d_s \in \mathbb{N}$, and producing for each one of them a stack of depth d_i , $i = 1, \dots, s$. In order to generate a stack, given A , d_i , and the special atom output, we first generated a batch containing a single rule, referred to as the stack’s *root*, with body from A and head output, and then we iteratively generated batches of average size b and interchangeably conflicting heads. Although this process does not explicitly determine the constructed target policy’s size or the average body size of its rules, we manipulated these attributes indirectly, by varying across policies the average batch size, flip count, number of stacks, and each stack’s root body size r . Namely, for the purposes of our experiments, r and b both ranged from 1 to 31 with a step of 2, while ℓ ranged from 1 to 31 with a step of 3, and s ranged from 1 to ℓ with a step of 2. For each assignment of values to r, b, ℓ, s , we generated 10 different target policies, to account (to some extent) for the various possible partitions of $\ell + 1$ to s positive integers. Lastly, in all above cases, A contained 20 atoms. All related metrics that were manipulated during the data generation stage are illustrated in an example policy in Figure 1.

We measured the predictive ability of each hypothesis policy against each target policy in P , and, in the case of intensional proxy coaches, against the corresponding learned model. We refer to these metrics as learning *performance* and learning *conformance*, respectively, with each of the two recording correct predictions, wrong predictions, abstentions, and accuracy, as discussed in Section 4.1. The fitness metric used in the evolutionary mechanism of the extensional proxy coach was as described in Section 3.2, and is summarized in Table 1.

		Offspring		
		<i>Correct</i>	<i>Abstain</i>	<i>Wrong</i>
Parent	<i>Correct</i>	0	-1	-1
	<i>Abstain</i>	1	0	-1
	<i>Wrong</i>	1	1	0

Table 1
Relative fitness metric.

B. Performance Results (Details)

Table 2 contains all results regarding the average final performance using all four proxy coaches, while in Tables 3 and 4 we present performance on evaluation set aggregated within each of the several groups of target policies we have produced. For evaluation purposes, we have also labeled the coaching set and kept its labeled part, so in what follows we report results on all three datasets (exemplar, evaluation and coaching).

Most target policy attributes do not seem to significantly affect performance when it comes to single-tree approaches with the exception of policies containing few flips or large rule batches. Regarding flips, this behavior is somewhat surprising since policies containing few

Set	Performance												Conformance												
	Exemplar				Coaching				Evaluation				Exemplar				Coaching				Evaluation				
	μ	Q1	Q2	Q3	μ	Q1	Q2	Q3	μ	Q1	Q2	Q3	μ	Q1	Q2	Q3	μ	Q1	Q2	Q3	μ	Q1	Q2	Q3	
E1	c	.96	.97	.99	.99	.93	.94	.96	.98	.93	.93	.96	.97	.96	.97	.99	.99	.97	.98	.99	1.0	.96	.97	.99	.99
	w	.00	.00	.00	.05	.00	.00	.02	.00	.00	.00	.02	.05	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
	a	.04	.01	.01	.03	.03	.00	.00	.02	.04	.01	.01	.03	.04	.01	.01	.03	.03	.00	.01	.02	.04	.01	.02	.03
E2	c	.96	.98	.99	.99	.93	.93	.96	.98	.93	.93	.96	.98	.96	.98	.99	.99	.96	.98	.99	1.0	.96	.97	.99	.99
	w	.03	.01	.01	.02	.06	.02	.04	.07	.06	.02	.04	.07	.03	.01	.01	.02	.03	.00	.01	.02	.03	.01	.01	.03
	a	.01	.00	.00	.00	.01	.00	.00	.00	.01	.00	.00	.00	.01	.00	.00	.00	.01	.00	.00	.00	.01	.00	.00	.00
E3	c	.85	.80	.87	.92	.84	.77	.87	.91	.84	.78	.86	.81	.87	.83	.89	.94	.87	.82	.89	.94	.86	.81	.88	.93
	w	.12	.06	.10	.18	.13	.06	.12	.19	.13	.06	.12	.20	.10	.05	.09	.15	.10	.05	.09	.15	.10	.05	.09	.16
	a	.03	.00	.00	.00	.03	.00	.00	.00	.03	.00	.00	.00	.03	.00	.00	.00	.03	.00	.00	.00	.03	.00	.00	.00
E4	c	.94	.92	.94	.97	.78	.73	.78	.82	.80	.76	.81	.86	-	-	-	-	-	-	-	-	-	-	-	-
	w	.06	.03	.06	.08	.22	.17	.21	.26	.19	.14	.19	.23	-	-	-	-	-	-	-	-	-	-	-	-
	a	.00	.00	.00	.00	.00	.00	.00	.00	.01	.00	.00	.00	-	-	-	-	-	-	-	-	-	-	-	-

Table 2

Final performance with respect to all used sets w.r.t. both the target policy (Accuracy) and the Proxy Coach (Conformance) (c: “correct”, w: “wrong”, a: “abstain”, μ : mean, Q1: 1st quantile, Q2: median, Q3: 3rd quantile).

flips are structurally simpler, which was expected to benefit coaching and requires, thus, further investigation. On the contrary, poorer performance on policies containing large rule batches was expected since larger batches impose, in general, a richer and more complex policy structure overall. As far as forests are concerned, with the exception of relatively large policies, where the corresponding hypotheses seem to be sufficiently accurate, correct predictions seem to remain unaffected by variations in target policy’s attributes. This unexpected efficiency on larger target policies is also subject to further investigation.

In Figure 5, we present two sets of single policies, each chosen from one of the groups defined in Section 4.1. Namely, on the left column (Figure 5a) each policy is the group’s median with respect to the policy attribute varied within that group, while on the right column (Figure 5b), each policy is the group’s median with respect to the total number of epochs required during coaching. In both columns, the four stripes correspond to experiments E1-E4, respectively, while, within each stripe, the first row corresponds to high-value groups and the second row to low-value ones.

C. Conformance Results (Details)

So far, we have discussed results regarding learning efficiency against the target theory (performance). We have also computed, whenever applicable, learning efficiency with respect to proxy coaches (conformance). Namely, *Conformance* values recorded how many of the predictions of the current hypothesis policy on the evaluation set \mathcal{E} were correct against the proxy coach, wrong against the proxy coach, or abstentions, and the accuracy of the definite predictions (i.e., the number of correct predictions over the number of correct or wrong predictions). As in Section 4.1, *Relative Size* values recorded the size of the current hypothesis policy (relative to

Set	Performance														
	Exemplar					Coaching					Evaluation				
	Size	Flips	Stacks	Batches	Width	Size	Flips	Stacks	Batches	Width	Size	Flips	Stacks	Batches	Width
E1	c	.99	.98	.98	.98	.98	.98	.98	.98	.98	.98	.98	.98	.98	.98
	w	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
	a	.01	.02	.02	.01	.12	.02	.02	.01	.01	.02	.00	.01	.02	.02
E2	c	.98	.98	.99	.99	.98	.98	.98	.98	.98	.98	.98	.98	.98	.98
	w	.02	.02	.01	.09	.01	.01	.07	.02	.02	.01	.03	.05	.04	.12
	a	.00	.00	.00	.00	.05	.00	.00	.00	.00	.00	.00	.00	.00	.00
E3	c	.91	.84	.87	.81	.86	.83	.86	.85	.86	.86	.90	.83	.87	.80
	w	.08	.14	.13	.06	.14	.14	.09	.16	.13	.06	.13	.19	.06	.16
	a	.01	.02	.00	.13	.02	.00	.03	.01	.00	.00	.01	.02	.00	.14
E4	c	.93	.86	.87	.91	.87	.86	.92	.86	.87	.85	.80	.73	.74	.79
	w	.07	.12	.12	.08	.12	.14	.07	.13	.11	.11	.20	.26	.24	.20
	a	.01	.02	.01	.01	.01	.01	.00	.01	.02	.04	.00	.01	.01	.01

Table 3

Average performance by group of target policies for each of the three datasets. (left: high-value group, right: low-value group, **bold**: lowest performance w.r.t. dataset and protocol).

Set	Conformance														
	Exemplar					Coaching					Evaluation				
	Size	Flips	Stacks	Batches	Width	Size	Flips	Stacks	Batches	Width	Size	Flips	Stacks	Batches	Width
E1	c	.99	.98	.98	.90	.98	.99	.88	.98	.98	.98	.99	.98	1.0	.91
	w	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00
	a	.01	.02	.02	.10	.02	.01	.12	.02	.02	.01	.02	.00	.09	.00
E2	c	.97	.98	.99	.90	.99	.99	.88	.98	.98	.98	.98	.98	1.0	.90
	w	.03	.02	.01	.10	.01	.01	.07	.02	.02	.01	.02	.02	.00	.10
	a	.00	.00	.00	.00	.00	.00	.05	.00	.00	.00	.00	.00	.00	.00
E3	c	.91	.85	.89	.82	.89	.86	.86	.87	.88	.87	.91	.86	.90	.83
	w	.07	.13	.11	.04	.09	.14	.06	.12	.12	.12	.08	.12	.10	.04
	a	.01	.02	.00	.13	.02	.00	.03	.01	.00	.01	.01	.02	.00	.13

Table 4

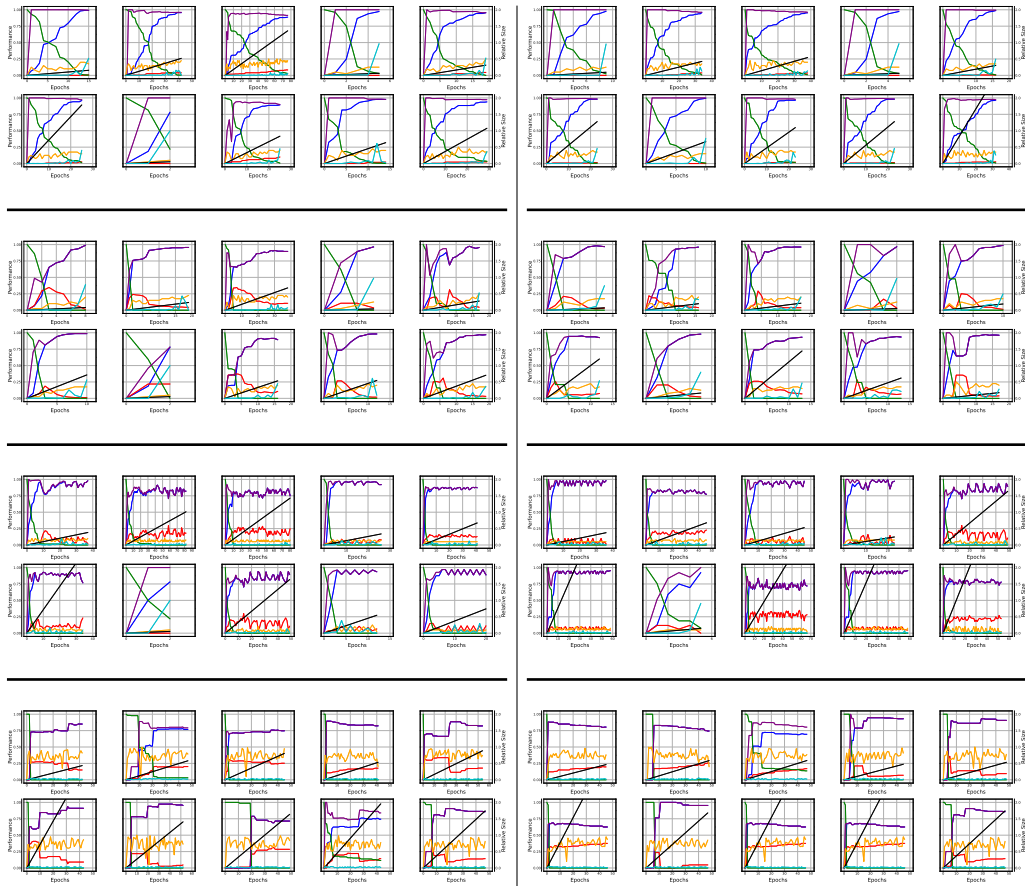
Average conformance by group of target policies for each of the three datasets. (left: high-value group, right: low-value group, **bold**: lowest performance w.r.t. dataset and protocol).

the size of p), and the size of the epoch (relative to the size of \mathcal{C}), the size of each piece of advice given (relative to the size of A).

All results regarding the above metrics are presented in Figure 6 (not applicable for E4). Overall, the trends observed for the three different advice protocols are similar to those presented in Figure 3. Nevertheless, one may observe that there are also some minor differences. To begin with, regarding the full path protocol, wrong predictions, in terms of conformance to the proxy coach’s ones, are zero. This is due to the advice-giving protocol being highly conservative on the returned advice which, as discussed in Section 3.1, is quite narrow in terms of coverage but highly accurate at the same time.

Another interesting fact is that, regardless of whether efficiency is measured against the target policy or the proxy coach, coaching under the forest protocol is persistently less efficient than the other two. This, again, could be interpreted as another hint for the inefficiency of

the underlying advice mining mechanism when it comes to forests or just as a side effect of a forest's relative opacity compared to single trees.



(a) Median policies per group, w.r.t. each policy attribute. (b) Median policies per group w.r.t. the total number of epochs.

Figure 5: Performance of the median policy of each group w.r.t. each group’s manipulated policy attribute (left column) and the total number of epochs within each group (right column). Within each column, the four stripes correspond to E1, E2, E3 and E4 respectively. Within each stripe, groups vary from left to right as follows: policy size, flip count, stack count, batches and width. Also, the top row of each stripe corresponds to high-value groups, while the bottom to low-value ones. Color coding is the same as in Figure 3.

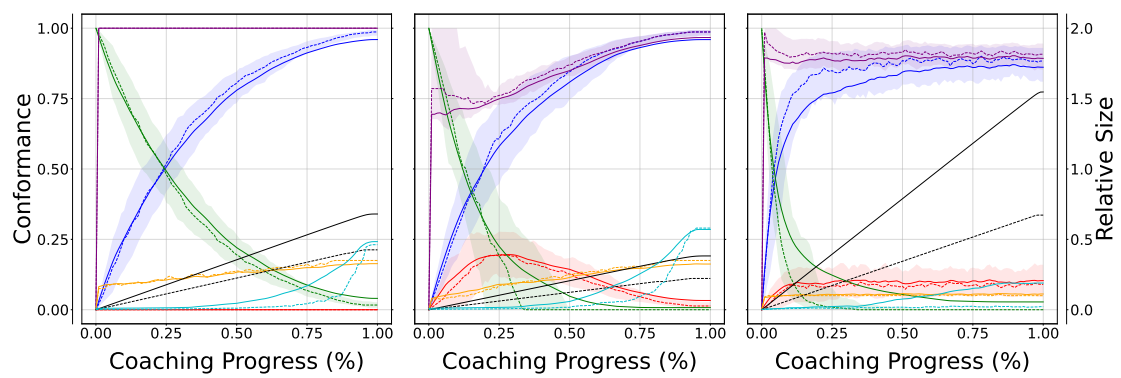


Figure 6: Results for experiments (E1)–(E4), shown from left to right. Conformance values show correct predictions (—), wrong predictions (—), abstentions (—), and accuracy (—) on the evaluation set, against the learned model. Relative Size values show the relative sizes of the current hypothesis policy (—), the epoch (—), and the given piece of advice (—). Results are aggregated across all target policies, with solid lines representing the mean values, dashed lines representing the median values, and shaded areas representing the $Q1$ to $Q3$ quartile interval.