An HMM-based real-time intervention methodology for a social robot supporting learning

Jauwairia Nasir, Mortadha Abderrahim, Barbara Bruno, Pierre Dillenbourg

Angaben zur Veröffentlichung / Publication details:

Nasir, Jauwairia, Mortadha Abderrahim, Barbara Bruno, and Pierre Dillenbourg. 2023. "An HMM-based real-time intervention methodology for a social robot supporting learning." In 2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 28-31 August 2023, Busan, Republic of Korea, 2204–11. Piscataway, NJ: IEEE. https://doi.org/10.1109/ro-man57019.2023.10309430.

An HMM-based Real-time Intervention Methodology for a Social Robot Supporting Learning

Jauwairia Nasir*§1, Mortadha Abderrahim*2, Barbara Bruno§3, and Pierre Dillenbourg2

***Abstract*— To make social robots effective in education, they need to be autonomous both in terms of assessing the student's engagement state as well as intervening effectively in soft real-time when necessary. Hidden Markov Model (HMM) is an interpretable machine learning technique for modeling temporal data that is commonly used post-hoc to analyse latent learning processes. In this paper, we contribute by proposing an HMM-based intervention methodology for assessing and classifying the state of the student as either productive or unproductive in soft real-time. The system identifies and tracks states and patterns not conducive to learning, and a robot intervention is triggered whenever a too-high non-productive engagement is detected. In a pilot study with 22 children, we evaluate this methodology in terms of both 1) the effectiveness of the interventions on the students' learning gains and on behaviors found conducive to learning, and 2) the students' perception of the robotic interventions. Results suggest that the robot interventions have a positive effect on the post-test scores relative to the baseline robot, although there isn't a significant difference in the learning gains. Moreover, interventions that try to induce reflective behaviors are most effective in inducing the required learning behavior, followed by communication-inducing interventions. Lastly, students' perception of intervention usefulness does not reflect their actual effectiveness.**

***Keywords*—human-robot interaction, real-time assessment, HMMs, engagement, educational robots, adaptive robots**

I. Introduction

Social robots are increasingly becoming a popular tool in educational settings [1], [2], [3]. They are used to support the acquisition of curricular knowledge [4], train transversal skills [5], [6], as well as promote healthy habits [7] or positive behaviours [8]. In most cases, real-time knowledge of the people's status and context is helpful, if not crucial, for the robot's proper assessment of the situation and identification of the course of action to follow. Concretely, in the context of robots supporting learning, this translates into designing robots capable of (i) understanding the students' state in real-time, in order to (ii) provide timely and effective feedback. Research on both capabilities, albeit very active, is still far from having reached a solution.

Concerning the first capability, several methodologies have been proposed to build such real-time systems for the assessment of students' state. Some solutions assess the student's skills on the basis of their performance in the learning activity, and then the robot would intervene accordingly [9]. Others move away from performance-based models to look for students' social and behavioral cues that reflect, for example, attention or engagement [10]. Concerning the second capability, a key challenge is that the robot should exhibit the appropriate level of social interactions to engage with the students without distracting them from the learning task. Therefore, the real-time methodology and the associated behavior of the social robot needs to be carefully designed to enhance the learning interaction.

Fig. 1: Students interacting with the learning activity with the robot

In this paper, targetting the two aforementioned abilities, we propose a model for real-time student state assessment and intervention, specifically designed for a social robot supporting learning in an open-ended collaborative activity (*JUSThink* [5], see Figure 1). The robot employs a Hidden Markov Model (HMM) to infer, based on the students' log actions and speech activity, the hidden states corresponding to unproductivity or productivity. With this information, the robot generates a score for the students that we term as *non-Productive Engagement* (nPE) score. The focus on *non-productiveness* in the score is because, in line with the second challenge highlighted above, the robot's first goal is to not to disturb those who are learning, i.e., focus on *unproductiveness* and react promptly whenever (and only when) it is detected. This score is then used by the robot to chose the intervention to perform. We evaluated the system in a pilot study involving 22 participants aged 9-12 years old.

*The first two authors contributed equally to this work.

§Majority of the work was conducted while the author was employed at CHILI Lab, EPFL

1Human-Centered Artificial Intelligence (HCAI) Lab, University of Augsburg, Germany

2Computer-Human Interaction in Learning and Instruction (CHILI) Lab, Swiss Federal Institute of Technology in Lausanne (EPFL), Switzerland

3Socially Assistive Robotics with Artificial Intelligence (SARAI) Lab, Karlsruhe Institute of Technology (KIT), Germany

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 765955.

II. Related Work

Different personalization approaches have been proposed in the literature that can be divided, depending on the features considered to model the students, into two main categories: 1) performance-based and 2) behavior-based.

Performance-based systems typically employ online Bayesian networks to evaluate students' skill mastering [11], [12], [13]. Bayesian networks model the relationship between observations, which are the student's answers, and skill assessment in probabilistic terms; the robot's interventions are then driven by the student's estimated skill level. For instance, in [11], a robot utilizes Bayesian networks to deliver personalized puzzle-solving lessons targeting the student's unmastered skills. Results show that participants who received personalized lessons outperformed those who received non-personalized ones. In addition to Bayesian networks, other performance-based models for robot tutors have also been employed. In [14], for example, the robot considers the students's performance in terms of accuracy and efficiency (the time it takes for the student to answer), and offers off-task breaks following three different strategies: at fixed times, when a significant increase in the performance is observed (as a reward), and when a significant decrease is observed (as a re-focus action). Results suggest that personalized timing strategies for breaks can increase students' learning gains.

A major limitation of performance-based system is the fact that students can learn from failures, as much and as well as they do from successes [15]. Thus, it makes sense that educational social robots also consider other indicators, especially in open-ended learning activities that employ the *learning by failing* paradigm. In [16], the authors integrate students' affective behaviors through an automatic facial expression analysis system measuring the students' valence and engagement. Combined with information about the status of the activity and the student's performance history, these features define the state space of an affective reinforcement learning algorithm, which is employed to personalize the robot's affective verbal and non-verbal response to each student. The proposed system helped the children learn new words in a second language and increased their positive valence compared to non-personalized interactions. Further, [17] considers speech activity as an approximate engagement metric in a robot tutoring activity where children are asked to use a think-aloud meta-cognitive learning strategy. Whenever a long period of silence is detected, the robot intervenes to encourage the student to speak. Students interacting with this robot version exhibited increased engagement and compliance, and higher immediate and persistent learning gains.

In this paper we employ a behavior-based student state assessment methodology. This choice is motivated by the fact that our learning activity is open-ended and exploratory by design, thus allowing for *learning by failing*. Specifically, we consider the students' actions on the learning activity and their speech behaviors and use them to model the student's *unproductiveness* in real-time via a Hidden Markov Model. Although relatively widely employed in learning analytics [18], [19], [20] and other real-time systems [21], [22] due to their ability to model latent processes interpretably, to the best of our knowledge, HMM-based intervention systems seems to be absent from literature in educational Human-Robot Interaction. Therefore, this is the gap area where the paper aims to contribute by designing and implementing such an intervention system in an educational social robot deployed in a school environment.

III. Background

We briefly describe the baseline *JUSThink* learning activity [5], as well as the outcomes that inspired the robot intervention methodology presented in this paper.

A. Activity

JUSThink is a collaborative learning platform which aims to provide intuitive knowledge about Minimum-Spanning Tree (MST) problems, thus improving children's computational thinking skills while also promoting collaboration within the team. The learning task presents the MST problem through a Swiss gold mining scenario. The robot, playing the role of the CEO of a gold-mining company, asks the students to connect the mines with railway tracks while spending as little money as possible on building them. Students collaborate to solve this open-ended problem without receiving direct guidance. The design of the learning task is such that the task can enforce better collaboration via choices such as *partial and complementary information* as well as *role switching* between the team members. Given this collaborative script, the team needs to communicate to get the complete information, build the solution, and agree on it before it is submitted for evaluation. Once submitted, the robot provides feedback on the proposed solution. The robot intervenes intermittently throughout the task to give motivational support through minimal verbal and non-verbal behaviors, for example, by encouraging the students when their solution is not optimal, addressing them by their names, suggesting to look at previous solutions after 5 wrong solutions have been submitted, etc. Additionally, the robot mediates and automates the entire activity by giving instructions and moving the activity from one stage to the next as required. More details on the activity can be found in [5].

B. Productive Engagement and Collaborative Learning Behavioral Profiles

In an earlier work using the aforementioned activity in a user study, a data-driven technique was adopted to identify a hidden link between student's behaviors and learning, that was termed as *Productive Engagement* (PE) [23]. Students' multimodal behavioral features (namely their actions within the learning activity, video features including affective state and gaze, and speech features) aggregated across the entire activity were considered for the analysis. This resulted in identifying three behavioral profiles, two exhibiting high learning gains (denoted as Type 1 gainers and Type 2 gainers), and a third profile exhibiting lower learning gains (non-gainers).

Comparing the resulting profiles from both a quantitative and qualitative point of view, it was observed in [24] that significant behavioral differences exist between the students who ended up learning and those who did not, specifically in terms of speech behaviors. Gainers tend to speak more as well as have interjecting speech without long pauses in their speech. At the same time, the two groups of gainers differ from one another in terms of the interplay between the problem-solving (PS) strategies adopted and affective states, with the type 1 gainers first exploring globally and then engaging in reflective actions (global PS strategy) while the type 2 gainers perform exploratory and reflective actions concurrently (local PS strategy). The former also exhibit a more expressive profile in terms of emotions (excitement and frustration) than the latter. As a result, the two gainer types are termed as the *Expressive Explorers* and the *Calm Tinkerers*, while the non-gainers are termed as the *Silent Wanderers* because of less communication and no clear PS strategy.

To further characterize how these multimodal behaviors evolve throughout the activity for the three groups of students, a Hidden Markov Model was employed in [25]. Each of the model yielded three hidden states for each of the three groups, where the states could be interpreted as either productive or unproductive based on the associated set of actions and speech behavior to each type. The analysis indicated that all students (intended as both gainers and non-gainers) have the highest probability of starting and staying in an unproductive state, characterized by lower levels of communication and fewer actions on the game interface. Furthermore, it revealed that *Expressive Explorers* and *Calm Tinkerers* do not exclusively adopt a single problem-solving strategy, as suggested by the aggregate profiles discussed above, but rather switch between the two strategies throughout the interaction. The *Silent Wanderers* also adopt similar problem-solving strategies, though with fewer reflection actions and higher probability of returning to the unproductive state.

In this paper, we first build a global HMM for all the students together to model the students' transitions between states. The sequence of these transitions could then help us identify and detect undesirable behaviors, in real-time, that should be mitigated by an intervention system, such as staying in the unproductive state or going back to it from productivity, as suggested above. To complement this assessment model, we then build a soft real-time intervention system and present a preliminary study with it in a school. To have a trade-off between accuracy and soft real-time execution, we consider the speech and log features only as input features for our model, as they represent the most discriminatory behaviors, as recommended by our previous findings.

IV. Methodology

A. Dataset

For training, we use the open-source temporal dataset [26] generated from the work described in the previous section. The dataset consists of around 20-25 minutes of interaction with the *JUSThink* learning activity per team (a total of 32 teams with students aged between 9 and 12), organized in windows of 10 seconds. Team level log actions, speech behavior, affective states, and gaze patterns in each window are reported in two formats, totaling 52 features. As highlighted above, in this analysis, we consider only a subset of the dataset as our training data: the log actions and speech features (for detailed definitions, please see [26]) in the non-incremental format, which means that the feature values at window i only refer to what happened in the $10s$ composing window i and do not depend on any previous windows.

The dataset also includes team-level learning and performance metrics, where performance is measured based on the cost of their final solution compared to the optimal one, whereas the learning gains are computed from the difference between the students' scores on their pre- and post-tests.

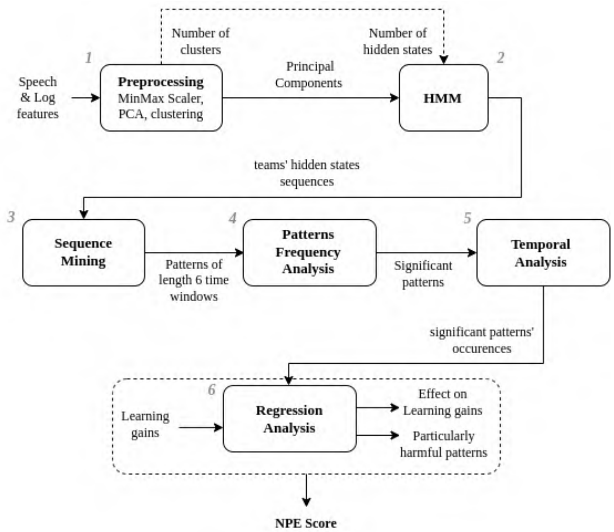

Fig. 2: System design steps

B. Design of the Intervention System

The design of the system consists of three main steps before the implementation stage for a real-time setting.

1) The Hidden Markov Model: First, using the training data, we employ an HMM to generate a multi-modal temporal behavioral profile for all the students together. This is represented by **steps 1 and 2** in the Figure 2. For this, speech and log features are first normalized using a MinMaxScaler. A behavior clustering step follows where a Principal Component Analysis is conducted, the reduced data is then clustered using the K-Means algorithm. The principal components are chosen based on the elbow method on the proportion of variance explained, resulting in 3 principal components explaining more than 0.8 of variance. The number of clusters, optimized based on the elbow method on inertia and the silhouette score, is three, and is considered a starting point for the number of hidden states of the HMM. We use the GMMHMM model provided by the hmmlearn library[1].

[1]hmmlearn is a set of algorithms for unsupervised learning and inference of HMM

Following this, the HMM is trained using the Expectation-Maximization algorithm on the teams' sequences with the number of hidden states set to the number of clusters. The Viterbi algorithm is then applied to the teams' sequences to determine at which state each observation is emitted. This permits constructing the set of observations emitted at each hidden state. Interpreting the significantly different features between pairs of states allows for characterizing the states in terms of the PS strategies and communication behavior. Specifically, three hidden states exist that could be interpreted as an unproductive state (state 0), a global PS strategy state (state 1), and a local PS strategy state (state 2).

2) Hidden State Sequential Analysis: The hidden state sequential analysis is represented in steps 3, 4, 5 and 6 in Figure 2. In order to pinpoint the differences between the learning process of those who learn and those who do not, in **step 2** we consider the sequence of the hidden states generated using the Viterbi algorithm for each of the teams. The analysis of these sequences could facilitate the identification of undesirable behaviors. Thus, in **step 3**, we start by applying the PrefixSpan[2] algorithm [27] on sequences of gainers and those of non-gainers separately to discover the frequent sequential patterns in each of them, with the minimum s-support set to 0.5, that is the minimum fraction of sequences that should contain the sequential pattern. Furthermore, we fix a time window of 60 seconds, i.e., we only consider patterns of length equal to 6. We observe that all possible permutations of length 6 exist in all non-gainer teams. We then investigate, in **step 4**, how the frequency of such patterns differs between gainers and non-gainers. A pattern is regarded as significant:

- if it exists exclusively in the sequences of gainers or in those of non-gainers
- if the difference in the distribution of its frequency is statistically significant (p-values < 0.01) between gainers and non-gainers.

The analysis suggests that 32 such significant patterns exist. All of them are more frequent or exist only in non-gainers' sequences. Most of these patterns can be described as unproductive patterns, which consist primarily of occurrences of the unproductive state such as: $[0,0,0,0,0,0]$, $[0,0,0,1,0,0]$, $[2,0,0,0,0,1]$, ... In **step 5**, a temporal analysis comparing the time when such patterns occur in the sequences of gainers and non-gainers shows that for the first 3 minutes, gainers and non gainers have a comparable number of occurrences of the significant patterns, whereas beyond that, the number of occurrences of such patterns is higher for the non-gainers. Thus, for the remaining of this analysis, we consider the number of occurrences of the significant patterns beyond this time mark. Next in **step 6**, we study the effect of the significant patterns on the learning gains. Considering the total number of occurrences of all significant patterns, we observe that this number is highly correlated with the team being a non-gainer team (*spearman correlation* = 0.69, *pvalue* = $1.21e^5$).We further train Ordinary Least Squares (OLS) models with the total number of occurrences of all significant patterns as the predictor and each of the learning gains (definitions in section V-B) as dependent variables. The global OLS models show that the total number of occurrences explains around 20% of the variance in each learning gain metric and has a negative, statistically significant coefficient. We additionally zoom into each pattern's effect on the learning gains. A regression analysis is conducted with the number of occurrences of each pattern as the predictor and each learning gain as the outcome. Eight patterns have a statistically significant coefficient for at least one of the learning gains, with all of these coefficients being negative. We refer to such patterns as the *particularly harmful patterns*. This analysis suggests that the occurrences of the significant patterns have a negative effect on the learning gains, with 8 of them being particularly harmful.

3) NPE score: In order for the intervention system to make use of the outcomes of the 6-steps-long analysis, we define the non-Productive Engagement score *nPE score*. This score characterizes the students as either being *productively engaged* or not. The *nPE score* can be considered as an alternate metric to the *PE score*, first defined to quantitatively characterize the concept of *productive engagement* in the chapter 6 of the PhD thesis [28]. To differentiate between the two types of quantification, the actions of a robot incorporating an *nPE score* based system are triggered by the presence of unproductive events while the actions of a robot incorporating a *PE score* based system are triggered by the absence of productive events. Concretely, the *nPE score* starts at zero and is updated as follows:

- For each occurrence of one of the 8 particularly harmful patterns, the score is incremented by the sum of the significant coefficients of its OLS models,
- For each occurrence of one of the other significant patterns, the score is incremented by the sum of the coefficients of the global OLS model,

thus giving more weight to the particularly harmful patterns. Since the coefficients are negative, as the correlation between these patterns and learning gain is negative, the score is never positive. This means that the smaller the score gets (or larger the absolute value of the score gets), the less *productively engaged* the team is gauged to be. The robot aims at *minimizing* the *absolute* value of the *nPE score* in real-time, i.e., bringing the *nPE score* towards zero. For that, the robot intervenes when the score of the team drops below a certain threshold τ_{nPE}. This is a time-dependent threshold generated as the weighted average of the average *nPE score* of the gainer teams at each time window and that of the non-gainer teams from the training data. The *nPE score*, along with the threshold and the intervention logic, is defined to penalize the high number of occurrences of the significant patterns while limiting the number of interventions for the gainer teams to avoid unnecessary distractions in the learning process. The time-dependent threshold also allows for adapting to the specificity of the different phases of the activity by being

[2] We use the implementation from pyspark MLlib

more permissive as the time evolves.

C. Robot Intervention Control Architecture

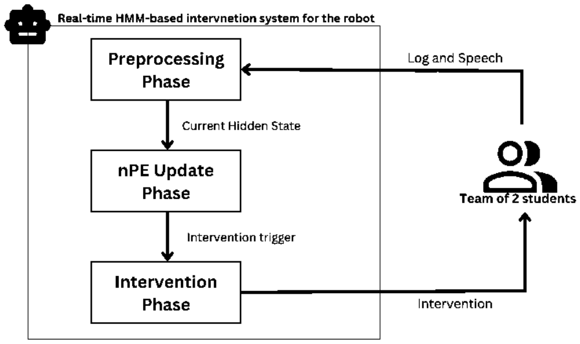

Fig. 3: Overview of the intervention control architecture for the robot

Now that we have a score that the robot can monitor in soft real-time, in this section we explain how the resulting real-time intervention system is implemented. The contol architecture can be split in three main phases: a *preprocessing phase* where the current behavioral features are preprocessed to get the current hidden state, the *nPE update phase* where the *nPE score* is updated based on the current pattern, and an intervention is triggered if needed. Finally, an *intervention phase*, where the intervention is executed and its weight is assessed based on its effectiveness. An overview of this architecture is shown in Figure 3.

1) Preprocessing phase: Algorithm 1 lists the steps executed in the preprocessing phase. Every 10 seconds, with the log and speech features as inputs, the current hidden state is generated for the team. The features are first normalized using a MinMax scaler (*line* 6). A PCA (7) is then applied to the normalized features where the resulting principal components are fed to the HMM's Viterbi algorithm (8) to generate the current hidden state. All these models are pre-trained on the dataset described above to be efficiently used in real-time.

Algorithm 1 Preprocessing Phase

1: *features* = raw features at time window *t*
2: *scaler*, *pca*, *hmm* = pre-trained models
3: *previous_features* = features after PCA at time window *t-1*
4: *current_state* = current hidden state
5: **procedure** PREPROCESS
6: *features* ← *scaler.transform (features)*
7: *current_features* ← *pca.transform(features)*
8: *current_state* ←*hmm.predict(previous_features, current_features)*
9: *previous_features* ← *current_features*
10: **return** *current_state*

2) NPE Score Update phase: Algorithm 2 lists the steps executed in the nPE score update phase. The current hidden state is enqueued in a First-In-First-Out buffer of size 6 (corresponding to 60 seconds) (9). Once the buffer is full, the pattern of 6 consecutive hidden states is compared against the significant patterns. If it is one of the significant patterns (10), the *nPE score* is updated as explained previously (11). Following that, if the updated score is below τ_{nPE} (12), an intervention is triggered(13).

Algorithm 2 nPE Score Update Phase

1: *t* = time window
2: *npe* = nPE score at time window *t-1*
3: *current_state* = hidden state at time window *t*
4: *threshold(t)* = threshold at time window *t*
5: *buffer* = states buffer of length 6
6: *significant_patterns* = list of significant patterns
7: *patterns_coefficients* = significant patterns coefficients
8: **procedure** NPE UPDATE AFTER 6TH TIME WINDOW
9: Put *current_state* in *buffer*.
10: **if** *buffer* in *significant_patterns* **then**
11: *npe* ← *npe* + *patterns_coefficients.get(buffer)*
12: **if** *npe* < *threshold(t)* **then**
13: Trigger an intervention

Algorithm 3 Intervention Phase

1: *t* = time window.
2: *interventions* = the interventions' contents and weights.
3: *npe* = nPE Score at time window t.
4: *npe_slope(t)* = nPE slope in the 12 time windows preceding *t*.
5: η = 0.2, learning rate
6: *last_intervention* =last intervention time window
7: **procedure** INTERVENTION EXECUTION
8: **if** *t-last_intervention* $>=$ 12 **then**
9: *exploration_probability* $\leftarrow 0.8 - 0.1 * (t\, div\, 30)$
10: *explore* ← *Bernoulli*(*exploration_probability*)
11: **if** *explore* **then**
12: *intervention* ← *random*(*interventions*)
13: **else**
14: *intervention* ← *argmax*(*interventions.weight*)
15: robot.intervene(*intervention.content*)
16: *npe* ← 0
17: *slope_before* ← *npe_slope(t)*
18: wait(12 time windows)
19: *slope_gain* = *slope_before*/*npe_slope(t)*
20: *intervention.weight* ← *intervention.weight* + η(*slope_gain* − *intervention.weight*)

3) Intervention phase: This phase (Algorithm 3) is executed only when an intervention is triggered and at least two minutes have passed since the last intervention (8). The pool of interventions consists of suggestions intended to increase the specific behaviors found to be lacking in the unproductive state. These include adding edges (exploration-inducing), looking at past solutions (reflection-inducing), and communicating with each other (communication-inducing).

We employed an exploration-exploitation policy where an intervention is chosen based on the outcome of a Bernoulli trial with a probability of exploration. The exploration probability is 0.8 at the start of the activity and decremented by 0.1 after every 5 minutes of interaction(9,10). When the Bernoulli trial is successful(11), the robot randomly chooses an intervention among all possible interventions(12); conversely, when the trial is a failure, the robot chooses the intervention with the highest weight(14). The weight for an intervention is updated based on the effect it has on the slope of the *nPE score*, by considering its value in the two minutes before and in the two minutes after the intervention as detailed in (19,20).

After the execution of an intervention by the robot, the score is reset to 0, which allows the robot to be more lenient after an intervention is made, thus offering the time for the team to adapt to its suggestion. Note that the buffer size of 60 seconds and the minimum timing between interventions of 2 minutes are set arbitrarily, aiming to have an interval neither too long nor too short between interventions.

V. Pilot Study

A. Participants

The pilot study, with the same platform but the robot additionally incorporated with the proposed intervention system, was conducted with 22 children aged 9 to 12 years, divided in teams of 2, at an international school in Switzerland. One team is omitted from the analysis due to technical problems during the experiment. The resulting dataset consists of 10 teams, totaling 20 children (7 females, M = 10.14, SD = 0.69; 13 males, M = 10.53, SD = 0.77). The learning activity lasted approx. 50 minutes for each team and was fully automated.

B. Evaluation Metrics

1) Pre- and Post- tests: Each test consists of 10 multiple-choice questions defined in a context other than Swiss gold mines. It assesses the student's understanding of 3 main concepts related to the MST problem: whether the graph is connected, whether a path spans the graph, and whether a path that spans the graph has minimum cost. Based on these tests, we define the following metrics:

- **pre- and post-test score**: ratio of questions answered correctly by each participant over the total.
- **Joint pre- and post-test scores**: ratio of questions that both team members answered correctly in the pre-test and post-test, respectively, over the total.

2) Learning gains (LGs): On the basis of the pre- and post-test scores, we compute the following learning metrics:

- **Absolute learning gain**: It is calculated as the difference between the post- and pre-test scores, divided by the maximum score that can be reached.
- **Relative learning gain**: It is the difference between the post- and pre-test scores, divided by the difference between the maximum score that can be reached and the pre-test score. The average of the two team members' absolute and relative learning gains gives the team's absolute and relative learning gains, respectively.
- **Joint absolute learning gain**: the knowledge the team members reached together, computed as the difference between the teams' joint post-test and joint pre-test scores.

3) Usefulness score: After each intervention, a pop-up window appearing on each participant's screen requests them to individually rate the robot's intervention as useful or not. The usefulness score can thus assume a value of 0 (both team members found the suggestion not useful), 0.5 (only one found it useful), or 1 (both found it useful). This indicates the students' perception of the intervention's usefulness.

C. Study Design

The robot welcomes the team and explains the task goal, followed by individual pre-tests. It then introduces the two game views and their functions. Then, the learning task, as explained in section III-A, starts. The robot intervenes throughout the activity following the proposed control architecture detailed in section IV-C. At the end, children take an individual post-test.

To have a reference, we consider the performance and learning gains of a subset of equal size of teams from the baseline experiment [5] where the robot only provided motivational support (see section III-A). Teams are matched on the propensity scores [29] computed on the pre-test scores, thus we estimate the effect of the robot interventions by accounting for the bias that could be induced by the knowledge the teams already have. Please note that our implementation is based on Robot Operating System (ROS).

Sepcifically, we investigate the following research questions: What is the impact of the proposed intervention system on the learning gains (**RQ-1**) as well as the relevant students' behaviors (**RQ-2**)? and how do students perceive the interventions delivered through the proposed intervention system (**RQ-3**)?

VI. Results and Discussion

A. Effect of the interventions on the learning gains

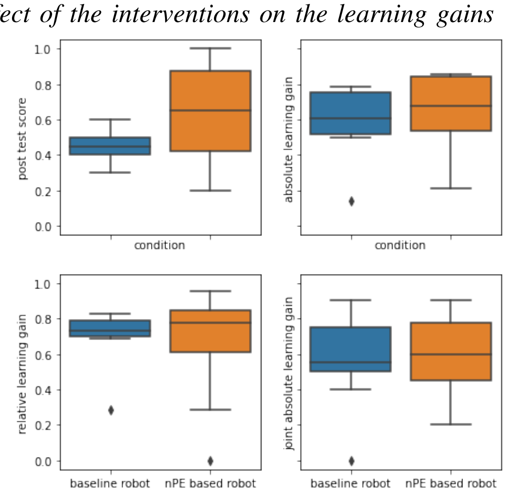

Fig. 4: Distribution of joint post-test score and learning gains by robot condition

For RQ-1, we consider the effect of the robot interventions on the learning gains and the post-test scores by comparing the values reached by teams undergoing this study to those exhibited by the reference teams, described above (section V-C). The teams from this pilot study have a median of 0.65 in the post-test score, 0.678 in absolute learning gain, 0.778 in relative learning gain, and 0.6 in joint absolute learning gain (see Figure 4 for reference). Although these median values are higher than those of the control teams (0.45, 0.607, 0.732, and 0.55, respectively), according to a Kruskal-Wallis test, only the difference in the post-test score is marginally significant (p = 0.097, 0.22, 0.649, and 0.402, respectively). It is interesting to note that the intervention seems to be increasing the variance, i.e. the intervention system seems to be beneficial to some and potentially detrimental to others.

Moreover, teams with the same number of interventions may have greatly dissimilar LGs, and conversely, as Figure 5 shows. This dissimilarity as well as the increase in variance in the LGs might be explained by the fact that interventions may not be equally effective across intervention types and teams.

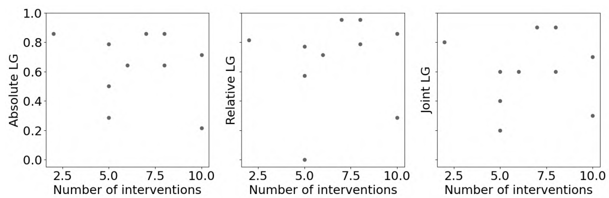

Fig. 5: Learning gains as a function of number of interventions

B. Effect of the interventions on student behaviours

For RQ-2, we compare the mean values of relevant behaviors before and after each intervention by analyzing features representative of those behaviors in a 2-minute time frame. We observe that 54.54% of the interventions lead to an increase in the teams' verbal behaviors (represented by *Speech_Activity* in the dataset [26]), and 50% increased in interjecting speech (*Speech_Overlap* in [26]). Moreover, 34.84% of the interventions resulted in a decrease in the long pauses in the speech (*Long_Pauses* in [26]). This is interesting as these behaviors are exactly the three contributing factors that define the *PE score* [28], where both speech quantity and interjecting speech had a positive effect on the *PE score* and the long pauses had a negative effect on it. Further, 51.51% of the interventions successfully induce the behaviors they were designed to induce. However, this impact varies between intervention types: only 40% of the *exploration-inducing* interventions increased the desired behavior, whereas this percentage reaches 57.69% for the *speech-inducing* interventions and 70% for the *reflection-inducing* ones.

Conclusively, the interventions do not only lead to an increase in the behaviors they are designed to stimulate but may also impact other behaviors. Their effectiveness, however, depends on the intervention type and the team receiving them. In fact, teams receiving the same number of interventions may exhibit dissimilar learning gains. For instance, considering the two teams receiving the highest

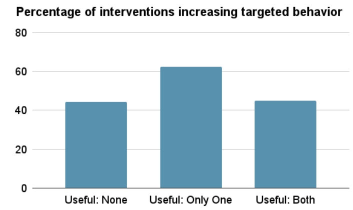

Fig. 6: Effect of usefulness on targeted behavior

number of interventions, we observe that only one of them reaches high learning gains. This may be explained by the difference in how effectively the interventions augment productive behaviors, particularly reflection behaviors.

C. Students' perception of the interventions

For RQ-3, we examine how the students perceive the interventions. 28.57% of the interventions were found not useful by both team members, 25.39% were considered useful by only one team member, whereas in 46.03% of the cases both team members found them useful. However, the usefulness, as perceived by the students, does not reflect the actual effectiveness of the interventions (see Figure 6). In fact, 44.44% of the interventions judged as not useful by both team members lead to an increase in the targeted features. The percentage is similar (44.83%) for the interventions that were found useful by both, whereas it reaches 62.5% for those that are found useful by only one member. Furthermore, when zooming in, speech activity, a behavior that is significantly higher in productive state, was effectively induced by 66.66% of the non-useful interventions compared to 43.75% of the semi-useful and 65.51% of the useful ones. Hence, the students' perception should not be the only metric to evaluate the usefulness of the robot's actions.

VII. Conclusion

In this paper, as our main contribution, we propose an HMM based real-time intervention system for a social robot to support students in an open-ended collaborative activity. To assess the students' states during the activity, the robot employs a Hidden Markov Model on their log and speech behaviors. The hidden states are interpreted as different states of unproductivity/productivity and problem-solving strategies. By analyzing the sequences of the hidden states, specific patterns are found to be harmful in terms of learning gains. The robot detects these patterns in real-time, generating the *non-Productive Engagement* (nPE) score. Whenever the *nPE score* drops beyond a time-dependent threshold, the robot performs an intervention to stimulate productive behaviors.

The pilot study, with 22 students, found that while the robot led to an insignificant increase in the learning gains, the interventions did stimulate the desired behaviors, as well as other behaviors indicative of *productive engagement*. Their effectiveness varies, with reflection-inducing interventions being the most effective. Furthermore, the actual effectiveness of the interventions does not necessarily align with the students' perception of their usefulness. These findings, while not definitive, offer insights for autonomous educational social robot interventions: 1) an HMM-based method to assess the dynamic process of learning via student behaviors in real-time is computationally effective, 2) there is a need to move away or complement the self-perception metrics and the post-hoc metric of learning gain with data-driven dynamic metrics (such as conducive to learning behaviors) in order to effectively evaluate pedagogical robots. Lastly, due to the small number of participants in our pilot study, further research with a larger group is necessary to confirm the effectiveness of the real-time intervention system.

References

[1] W. Johal, "Research Trends in Social Robots for Learning," *Current Robotics Reports*, vol. 1, pp. 1–9, 2020.

[2] S. Said, S. Al Kork, T. Beyrouthy, and K. Youssef, "Social Robotics in Education A Survey on Recent Studies and Applications," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 18, p. 67, 2023.

[3] W. Johal, T. Belpaeme, and M. Chetouani, "Editorial: Robots for learning," *Frontiers in Robotics and AI*, vol. 9, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frobt.2022.1050658

[4] E. A. Björling, B. Louie, P. Wiesmann, and A. C. Kuo, "Engaging english language learners as cultural informants in the design of a social robot for education," *Multimodal Technologies and Interaction*, vol. 5, no. 7, p. 35, 2021.

[5] J. Nasir, U. Norman, B. Bruno, and P. Dillenbourg, "When positive perception of the robot has no effect on learning," in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020, pp. 313–320.

[6] P. Alves-Oliveira, P. Arriaga, A. Paiva, and G. Hoffman, "Children as robot designers," in *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 2021, pp. 399–408.

[7] P. Prabha, S. Sasidharan, D. Pasupuleti, A. Das, G. Manikutty, and R. Sharma, "A minimalist social robot platform for promoting positive behavior change among children," in *ACM SIGGRAPH 2022 Educator's Forum*, 2022, pp. 1–2.

[8] E. Sanoubari, J. E. Muñoz Cardona, H. Mahdi, J. E. Young, A. Houston, and K. Dautenhahn, "Robots, bullies and stories: A remote co-design study with children," in *Interaction Design and Children*, 2021, pp. 171–182.

[9] A. Ramachandran, S. S. Sebo, and B. Scassellati, "Personalized Robot Tutoring using the Assistive Tutor POMDP (AT-POMDP)," *Proceedings of The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1–8, 2019.

[10] G. Gordon, S. Spaulding, J. K. Westlund, J. J. Lee, L. Plummer, M. Martinez, M. Das, and C. Breazeal, "Affective personalization of a social robot tutor for children's second language skills," *Proceedings of the 30th Conference on Artificial Intelligence (AAAI 2016)*, no. 2011, pp. 3951–3957, 2016.

[11] D. Leyzberg, S. Spaulding, and B. Scassellati, "Personalizing robot tutors to individuals' learning differences," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 423–430. [Online]. Available: https://doi.org/10.1145/2559636.2559671

[12] G. Gordon and C. Breazeal, "Bayesian active learning-based robot tutor for children's word-reading skills," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, Feb. 2015. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/9376

[13] T. Schodde, K. Bergmann, and S. Kopp, "Adaptive robot language tutoring based on bayesian knowledge tracing and predictive decision-making," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 128–136. [Online]. Available: https://doi.org/10.1145/2909824.3020222

[14] A. Ramachandran, C.-M. Huang, and B. Scassellati, "Give me a break! personalized timing strategies to promote learning in robot-child tutoring," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 146–155. [Online]. Available: https://doi.org/10.1145/2909824.3020209

[15] M. Kapur and C. K. Kinzer, "Productive failure in cscl groups," *International Journal of Computer-Supported Collaborative Learning*, vol. 4, no. 1, pp. 21–46, 03 2009, copyright - International Society of the Learning Sciences, Inc.; Springer Science + Business Media, LLC 2009. [Online]. Available: https://www.proquest.com/scholarly-journals/productive-failure-cscl-groups/docview/234900688/se-2

[16] G. Gordon, S. Spaulding, J. Kory Westlund, J. J. Lee, L. Plummer, M. Martinez, M. Das, and C. Breazeal, "Affective personalization of a social robot tutor for children's second language skills," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/9914

[17] A. Ramachandran, C.-M. Huang, E. Gartland, and B. Scassellati, "Thinking aloud with a tutoring robot to enhance learning," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 59–68. [Online]. Available: https://doi.org/10.1145/3171221.3171250

[18] K. Sharma, Z. Papamitsiou, J. K. Olsen, and M. Giannakos, "Predicting learners' effortful behaviour in adaptive assessment using multimodal data," in *Proceedings of the Tenth International Conference on Learning Analytics Knowledge*, ser. LAK '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 480–489. [Online]. Available: https://doi.org/10.1145/3375462.3375498

[19] Y. Xiao, Q. He, B. Veldkamp, and H. Liu, "Exploring latent states of problem-solving competence using hidden markov model on process data," *Journal of Computer Assisted Learning*, vol. 37, no. 5, pp. 1232–1247, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jcal.12559

[20] H. Jeong, A. Gupta, R. Roscoe, J. Wagster, G. Biswas, and D. Schwartz, "Using hidden markov models to characterize student behaviors in learning-by-teaching environments," in *Intelligent Tutoring Systems*, B. P. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 614–625.

[21] N. Nguyen-Duc-Thanh, S. Lee, and D. Kim, "Two-stage hidden markov model in gesture recognition for human robot interaction," *International Journal of Advanced Robotic Systems*, vol. 9, no. 2, p. 39, 2012. [Online]. Available: https://doi.org/10.5772/50204

[22] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann, "A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models," *IEEE Transactions on Robotics*, vol. 21, no. 1, pp. 47–57, 2005.

[23] J. Nasir, B. Bruno, M. Chetouani, and P. Dillenbourg, "What if social robots look for productive engagement?: Automated assessment of goal-centric engagement in learning applications," *International Journal of Social Robotics*, vol. 14, 01 2022.

[24] J. Nasir, A. Kothiyal, B. Bruno, and P. Dillenbourg, "Many are the ways to learn identifying multi-modal behavioral profiles of collaborative learning in constructivist activities," *International Journal of Computer-Supported Collaborative Learning*, 2021.

[25] J. Nasir, M. Abderrahim, A. Kothiyal, and P. Dillenbourg, "Temporal pathways to learning: How learning emerges in an open-ended collaborative activity." *Computers and Education: Artificial Intelligence*, vol. 3, p. 100093, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666920X22000480

[26] J. Nasir, B. Bruno, and P. Dillenbourg, "PE-HRI-temporal: A Multimodal Temporal Dataset in a robot mediated Collaborative Educational Setting," Nov. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5576058

[27] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth," 02 2001, pp. 215–224.

[28] J. Nasir, "Introducing Productive Engagement for Social Robots Supporting Learning," p. 250, 2022. [Online]. Available: http://infoscience.epfl.ch/record/297523

[29] P. R. ROSENBAUM and D. B. RUBIN, "The central role of the propensity score in observational studies for causal effects," *Biometrika*, vol. 70, no. 1, pp. 41–55, 04 1983. [Online]. Available: https://doi.org/10.1093/biomet/70.1.41