COVES: A Cognitive-Affective Deep Model that Personalizes Math Problem Difficulty in Real Time and Improves Student Engagement with an Online Tutor

Hao Yu
haoyu@bu.edu
Boston University
Boston, MA, United States

Danielle A. Allessio
allessio@umass.edu
University of Massachusetts Amherst
Amherst, MA, United States

Will Lee
williamlee@cs.umass.edu
University of Massachusetts Amherst
Amherst, MA, United States

William Rebelsky
wrebelsky@umass.edu
University of Massachusetts Amherst
Amherst, MA, United States

Frank Sylvia
fsylvia13@gmail.com
University of Massachusetts Amherst
Amherst, MA, United States

Tom Murray
tmurray@cs.umass.edu
University of Massachusetts Amherst
Amherst, MA, United States

John J. Magee
jmagee@clarku.edu
Clark University
Worcester, MA, United States

Ivon Arroyo
ivon@cs.umass.edu
University of Massachusetts Amherst
Amherst, MA, United States

Beverly P. Woolf
bev@cs.umass.edu
University of Massachusetts Amherst
Amherst, MA, United States

Sarah Adel Bargal
sarah.bargal@georgetown.edu
Georgetown University
Washington, D.C., United States

Margrit Betke
betke@bu.edu
Boston University
Boston, MA, United States

ABSTRACT

A key to personalized online learning is presenting content at an appropriate difficulty level; content that is too difficult can cause frustration and content that is too easy may result in boredom. Appropriate content can improve students' engagement and learning outcome. In this research, we propose a computer vision enhanced problem selector (COVES), a deep learning model to select a personalized difficulty level for each student. A combination of visual information and traditional log data is used to predict student-problem interactions, which are then used to guide problem difficulty selection in real time. COVES was trained on a dataset of fifty-one sixth-grade students interacting with the online math tutor MathSpring. Once COVES was integrated into the tutor, its effectiveness was tested with twenty-two seventh-grade students in controlled experiments. Students who received problems at an appropriate difficulty level, based on real-time predictions of their performance, demonstrated improved engagement with the math tutor. Results indicate that COVES leads to higher mastery of math concepts, better timing, and higher scores, thus providing a positive learning experience for the participants.

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10.
https://doi.org/10.1145/3581783.3613965

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; • **Computing methodologies** → **Computer vision**; • **Applied computing** → **Education**.

KEYWORDS

Deep machine learning, multi-modal fusion, problem difficulty selector, intelligent tutoring system, learning outcome, facial expression recognition, log data, real-time prediction.

ACM Reference Format:
Hao Yu, Danielle A. Allessio, Will Lee, William Rebelsky, Frank Sylvia, Tom Murray, John J. Magee, Ivon Arroyo, Beverly P. Woolf, Sarah Adel Bargal, and Margrit Betke. 2023. COVES: A Cognitive-Affective Deep Model that Personalizes Math Problem Difficulty in Real Time and Improves Student Engagement with an Online Tutor. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3581783.3613965

1 INTRODUCTION

Learners need challenging tasks to promote maximum cognitive growth; they perform best when tasks are just beyond their capabilities or out of their ability range [13, 22]. To ensure that students' mastery level continues to rise, teachers or "more knowledgeable others" should provide ongoing, scaffolded support and new opportunities so learners work slightly beyond their current skill level. To describe what learners can achieve with guidance and encouragement from a skilled partner and what they can do without help, Vygotsky [22] defined the "zone of proximal development" (ZPD). The term "proximal" refers to exposing learners to skills that have

a possibility of being mastered, and the term "zone" to a set of activities that are most beneficial, when tasks are just beyond learners' capabilities. Vygotsky noted that good teachers should not present material that is too difficult and support students to succeed by solving problems with help [4].

Within the context of intelligent tutoring systems, previous algorithms that select appropriate learning activities have mostly been based on heuristics. For instance, the "Effort Based Tutoring" problem selection algorithm [1] for MathSpring, an intelligent tutor for fourth-grade and above (mathspring.org), attempts to keep students in the ZPD with a rule-based algorithm that reasons about students' recent actions, in terms of attempts, hints and time spent [15]. We, however, questioned whether the tutor's response might be improved if it interpreted the face and gesture of its online students. Recent advances in computer vision [18] enable tutoring systems to observe students' facial features, Figure 1. Students express complex cognitive-affective states as they interact with online systems that can enrich the bandwidth of communication between tutor and learners, with the potential to better understand learning.

Our main research questions are: Can analysis of students' facial expressions with computer vision improve on the state-of-the-art in educational content selection? How can we design a machine learning model that improves the choice of the next math problem by looking at the students' facial features, and thus optimize student engagement? We address these questions using the intelligent tutor MathSpring (its interface is shown in Figure 2) as a real-time testing platform for our proposed machine learning model. We call our model COVES for computer vision enhanced problem selector. It is a deep learning model that personalizes difficulty levels of math activities by attempting to optimize each student's engagement state. It takes on the role of a "knowledgeable other" (e.g., a teacher) who has knowledge about learners and what might be best for them [13]. The model takes as inputs the student's video feed and information about the student's current online problem solving performance. The model estimates how the student will perform on the next problem and uses this to predict the difficulty level of the next math problem that the tutor should select. Based on COVES, the MathSpring tutor selects the math problem in real time that is best suited for the student at that point during the online interaction. The goal is for students to remain in a positive/engaged learning state (solving problems correctly, after hints, or after a single incorrect attempt) and to keep them away from negative behavior/disengagement (quick-guessing, not reading, skipping problems, or giving up), see Effort Chart, Figure 1, right.

The main challenge we face then is how to accurately predict a student's problem solving performance or outcome on the next problem. Previous methods of automatic problem solving outcome prediction in intelligent tutors have focused on analyzing student videos using facial expression recognition techniques [10, 18, 19]. While prior work has shown the potential for accurate outcome prediction based on visual data alone, these methods lack the context and complementary information provided by student log data, which is commonly used by human teachers to interpret student behavior. In this work, we propose a novel multimodal analysis approach that combines state-of-the-art visual affective analysis with student log data (i.e., information about how the students performed previously and the difficulty levels of the problems) using an attention-based fusion module. To the best of our knowledge, we are the first to integrate these two sources of information to improve the accuracy of student problem solving outcome prediction.

We trained our outcome prediction model on the MathSpring Children Dataset [18], a dataset of fifty-one sixth-grade students interacting with the MathSpring intelligent tutor. We integrated COVES into MathSpring and conducted a real-time classroom experiment with twenty-two seventh-grade students. We randomly divided participants into "treatment" and "control" groups, where learners in the treatment group used MathSpring with the COVES math problem selector, and learners in the control group used MathSpring's legacy "Effort Based Tutoring" problem selector [1]. Thus, our work "closes the loop" for computer-vision-based real-time selection of problem difficulty in tutoring systems; that is, computer vision is used not only for post-experiment analysis of student performance, as in previous work [10, 18, 19], but, to the best of our knowledge, for the first time, incorporated into an online tutor for real-time decision making by the tutor.

To protect student privacy, COVES conducts the analysis of student videos only locally, in the front end of the web-based software. No student videos are stored on the web. To achieve real-time prediction locally with limited computation resources, we applied several optimization techniques: We downsampled the video, used model quantization to speed up the algorithm, and reduced the model size.

In summary, this paper makes three main contributions to the literature:

- We propose COVES, a novel deep learning model for selecting personalized difficulty levels for each student in an intelligent tutor, based on predicting student problem solving outcomes.
- We introduce an attention-based deep fusion model that combines visual and non-visual data for improving the performance of problem solving outcome prediction.
- We demonstrate the real-time incorporation of our COVES model into an online tutor and conduct classroom experiments using the COVES-enhanced tutor. Statistically significant results indicate that COVES leads to higher mastery of math concepts, better timing, and higher scores, thus providing a positive learning experience for the participants.

2 RELATED WORK

2.1 Difficulty Selection in Intelligent Tutors

Presenting content at an appropriate difficulty level is a key goal of intelligent tutoring systems. Some existing implementations are based on deterministic selection rules [1, 5, 20]. For example, Sampayo-Vargas et al. [20] adaptively adjusted content difficulty level following simple rules such as decreasing one level of difficulty for three consecutive incorrect answers. CIRCSIM-Tutor [5] considered both students' self-report and prior performance (student's cumulative score on past problems). More recent approaches have employed reinforcement learning to model the difficulty selection process [12, 17, 21, 23, 24]. They used multi-armed bandits framework to learn an exploration-exploitation policy to select difficulty levels in order to maximize learning gains. While these stochastic

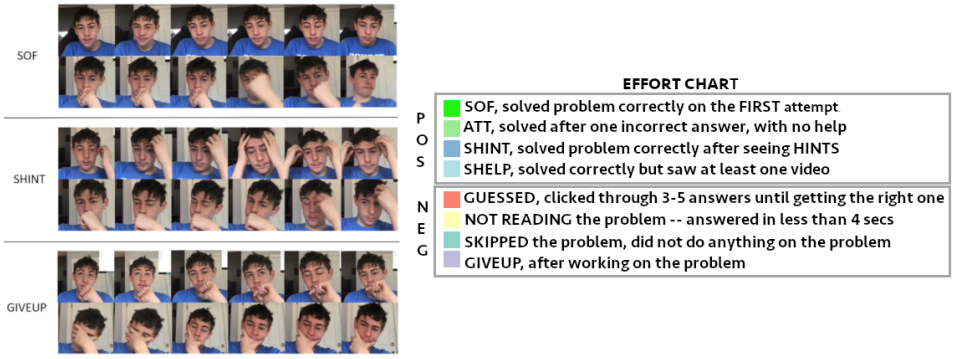

Figure 1: Example face-cropped images showing the evolution of student expressions and gestures with the corresponding problem sovling outcomes. The top four rows indicate positive learning outcome and the bottom two rows indicate negative learning outcome. In the top two rows the student solved the problem on the first attempt (SOF), in rows 3-4 the student solved the problem with hints (SHINT) and in the bottom two rows the student tried but ultimately skipped the problem (GIVEUP).

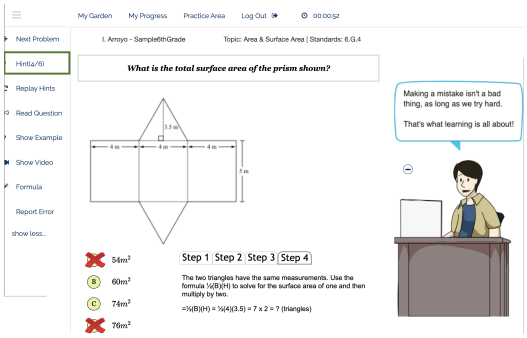

Figure 2: The Practice Area interface of an existing tutor, MathSpring. Hints are available from the "Hints" button on the left, which are supplemented with audio for any text displayed. Worked-out examples, tutorial videos, and formulas are also accessible. Jake (right) is a learning companion that talks to students about the importance of effort and perseverance, in the precise moment that they make mistakes.

approaches are robust to noise compared to deterministic ones, they can be difficult to tune in practice.

The legacy MathSpring "Effort Based Tutoring" (EBT) problem selector decides to increase/decrease content difficulty based on previous student behavior (correctness, hints requested, and time spent on the last math problem). For example, disengaged behavior, see Figure 1 GIVEUP, produces a reduction in problem difficulty, based on the assumption that if students are not working hard enough on the current problem, they probably will not work hard on a similar or more difficult problem. However, once EBT decides to increase problem difficulty, which one (of all) more difficult problems to select relies on a simple function that always selects the problem at the 50th percentile among a problem-difficulty-sorted list of available problems (of math problems harder than the one on which the student just worked). By contrast, in the new COVES algorithm, when the EBT problem selector decides to increase the difficulty, instead of directly selecting the one at the 50th percentile, the algorithm selects a difficulty level using its deep learning model that attempts to maximize the probability of a positive problem solving outcome (SHINT–solved with hints; SOF–solved on first try; ATT–solved after one incorrect attempt).

2.2 Problem Solving Outcome Prediction

In prior research, the problem of predicting student learning outcomes in intelligent tutors has been studied [10, 18, 19]. Joshi et al. [10] first introduced a labeled video dataset of student interactions with MathSpring and predicted outcomes using traditional facial affect signals such as head pose, gaze, and facial action units (AUs). Ruiz et al. [19] augmented the video dataset and developed a transfer learning approach to leverage a deep affect representation for outcome prediction, achieving state-of-the-art performance. Ruiz and Yu et al. [18] further extended the analysis by including student engagement prediction, enabling exploration of how engagement and learning outcomes correlate. Different from the previous research that considers video information only, the research described in this paper combines the state-of-the-art visual representation with log data to predict problem solving outcome.

3 METHODS

The proposed COVES math problem selector replaces the legacy "Effort Based Tutoring" (EBT) problem selector, discussed above. It establishes the desired difficulty level for the next problem that has the most likelihood of keeping students engaged (or re-engaged). MathSpring then chooses the problem with the closest difficulty available to this "desired" problem level. This is challenging given all the variables that might impact student performance and motivation, see Fig 3 and Section 3.1. It is also difficult to evaluate this multi-dimensional process because real-time conditions keep changing in an adaptive tutor and evaluative reference points evolve.

3.1 Complexity of Student Data in Tutor Session

As an indication of the complexity of data available at the moment of making real-time decisions about selecting next problem difficulty, we present Figure 3. This illustration demonstrates just some of the

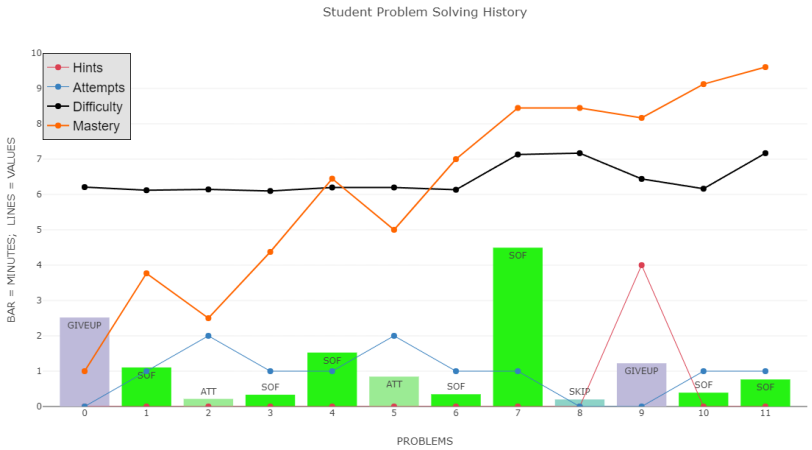

Figure 3: Complexity in problem selection. This visualization shows the time (vertical axis, minutes) that a real student spent working on twelve problems (horizontal axis), as well as the corresponding problem difficulty (black) and mastery levels (orange) (vertical axis, level). Student mastery (orange) kept increasing showing that the student was learning. Problem difficulty level (black) increased after a sequence of positive outcomes but then decreased since the student skipped problem 8 and gave up on problem 9. The problem difficulty only increases again after the student solves problem 10 on the first attempt.

degrees of freedom at work in this complex environment. It shows one student's actual mastery level (orange), in relation to the choice of problem difficulty (black). The tool shows a student's solution for twelve problems (horizontal axis) about one mathematics topic and minutes spent (vertical axis) along with attempts, performance (mastery) and problem difficulty values.

The graphic in Figure 3 shows that student mastery (orange) kept increasing, thus, the student appears to be learning. Problem difficulty level (black) increased after a sequence of positive outcomes but then decreased since the student skipped a problem and gave up on another problem. The problem difficulty only increases again after the student solves a problem on the first attempt.

3.2 The Proposed COVES Model

Given the large number of variables in selecting problem difficulty, Figure 3, we used computer vision and machine learning (ML) to predict student learning and then select the next problem. We developed a deep learning based model that predicts the possible problem solving outcome given a student's previous learning behavior and the candidate difficulty level of the next problem. By analyzing a student's learning log data as well as current facial expressions and gestures, COVES predicts the student's problem solving outcome as either positive or negative, Figure 1, right. The problem difficulty level that yields a positive outcome will be selected based on prediction results and a problem with a specific difficulty level presented as the next problem to be solved.

Suppose for a student who just solved the kth problem in a learning session, we have the student's learning log data (m_k, d_k) where m_k is how well the student has mastered the exercises so far and d_k is the difficulty level of the kth problem, the candidate difficulty level for the $(k+1)$th problem d_{k+1}, and a video with T frames $X = (X_1, X_2, \ldots, X_T)$ capturing the faces and gestures of the student while solving the kth problem. For clarity, we denote all the non-visual data as $D = (m_k, d_k, d_{k+1})$. m_k, d_k, d_{k+1} are all floating point numbers ranging from 0 to 1, and the larger number represents the higher level of mastery or difficulty. MathSpring calculates the mastery level based on the previous learning activities of the student, which involves factors such as the number of problems attempted for a specific topic, the number of problems solved correctly, the number of mistakes made, and the number of problems solved with some assistance. The difficulty level of a problem is estimated from the three independent sources of evidence of students' effort to solve a problem: the correctness in terms of the number of attempts to solve a problem, the amount of time spent on the problem, and the amount of help required or requested to solve the problem correctly [1].

Given X and D (for frames and non-visual data), our task is to predict the possible problem solving outcome y as either positive or negative, if the problem with difficulty d_{k+1} is presented as the next problem. COVES consists of three main components (Figure 4), a video analysis module that encodes the visual information of students (left), a fusion module that combines the visual input with non-visual data to predict problem solving outcome (right), and a difficulty level selection module (bottom, pink).

3.2.1 Video Representation Learning. Following Ruiz et al. [19], we adopted three deep learning networks to encode and analyze the videos of students' who gave permission for videos of their faces and gestures while they solve problems to be published, Figure 4. As facial expressions and gestures are important cues for inferring outcomes (Figure 1), an affect network was trained using in-the-wild images for facial expression recognition and leverage transfer learning to learn an affect representation for each frame of student videos $\rho(X_i), i = 1, \ldots, T$ (Figure 4, magenta). Meanwhile, per-frame facial Action Unit (AU) presence and intensity, gaze direction, and head pose were extracted using a facial analysis network, denoted as $\psi(X_i), i = 1, \ldots, T$ (Figure 4, green). The outputs of the two networks are concatenated as the final representation for each frame and used in a unidirectional 2-layer long short-term memory

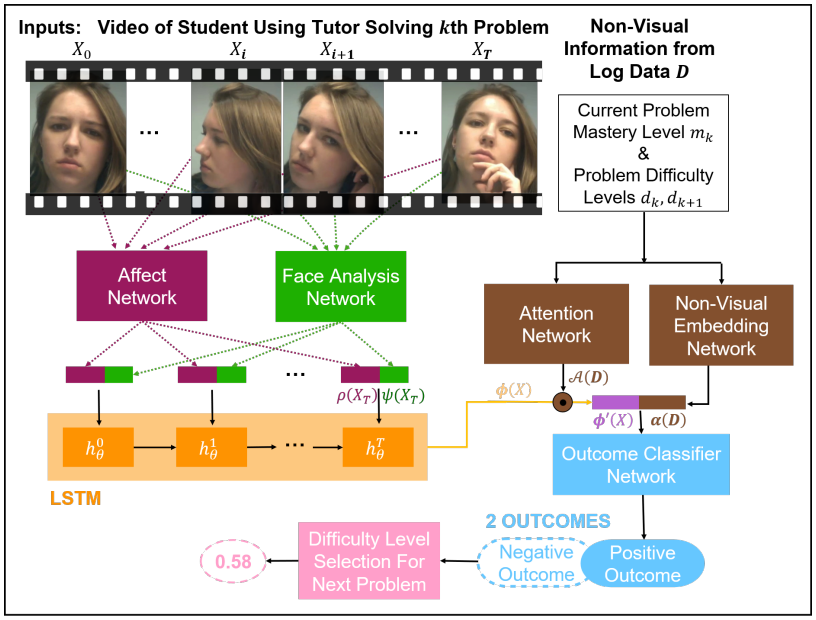

Figure 4: The computer vision enhanced problem selector (COVES) model based on problem solving outcome prediction. The model consists of three main components, a video analysis module that encodes the video of students using the tutor (left), a fusion module that combines the video information with non-visual information to predict problem solving outcome (right), and a difficulty level selection module (bottom, pink). The student image is of a MathSpring student who gave permission to use their video for analysis and publication.

(LSTM) network [9] to process them frame by frame. The LSTM network exploits the temporal patterns in the video and finally generates a video representation $\phi(X)$ (Figure 4, orange).

3.2.2 Fusion Module. Our methodological contribution here is to use the affective video embeddings, computed as described in Section 3.2.1, and combine them with embeddings computed for the non-visual data in a fusion model. We first encode the non-visual data using a non-visual embedding network, consisting of a linear embedding layer α that produces a non-visual embedding $\alpha(D)$ with the same size as the video embedding $\phi(X)$. Meanwhile, an attention network [2] is learned from the non-visual data to highlight the salient regions in the video representation (Figure 4, brown). Concretely, the attention module takes D as input to infer the attention $\mathcal{A}(D)$ through a fully connected layer, where $\mathcal{A}(D)$ has the same dimensions as the video representation $\phi(X)$. We use the same dimensions so that we can fuse the two types of embeddings, based on visual and non-visual data, in a multiplicative way. More specifically, to make the sum of the attention of each feature in the video embedding $\phi(X)$ equal to one, we first normalize the attention using the softmax function. The attention-boosted video representation is then

$$\phi'(X) = \text{softmax}(\mathcal{A}(D)) \odot \phi(X),$$

where $\odot$ denotes element-wise multiplication (Figure 4, lilac). We then concatenate the attention-boosted video representation $\phi'(X)$ and the non-visual embedding $\alpha(D)$ and use a linear outcome classifier network C to predict problem solving outcome $y = C(\phi'(X) \oplus \alpha(D))$ (Figure 4, blue). The reason to not just simply use $\phi'(X)$ but also $\alpha(D)$ as the input to the classifier is that we want to give the classifier direct access to the non-visual data embeddings.

3.2.3 Difficulty Level Selection Module. Once we have a model that can predict the problem solving outcome y of a student given any difficulty level d_{k+1}, we can use it to find an optimal difficulty level that leads to a positive learning outcome for the next problem (Figure 4, pink). Given a set of candidate difficulty levels for the next problem $d_{k+1}^{(1)}, d_{k+1}^{(2)}, \ldots, d_{k+1}^{(m)}$, we can run the model for each $d_{k+1}^{(i)}$ to obtain the corresponding problem solving outcome $y^{(i)} (i \in [1, m])$ and select one with a positive outcome. For efficiency, we perform a binary search on a sorted list of candidate difficulty levels. We describe the detailed process in the Algorithm "Difficulty Level Selection for Next Problem" on page 6.

3.2.4 Training of Deep Fusion Model.

Dataset. We trained the outcome prediction model on MathSpring Children Dataset [18], which consists of 968 videos of fifty-one sixth-grade students interacting with MathSpring as well as the outcome annotations and the log data. We retrieved students' mastery levels and problem difficulty levels from the log data. Two consecutive problem-solving interactions are a data sample. Overall, 793 data samples have been generated for training.

Loss Function. To train the model, we adopted two cross-entropy losses, $\mathcal{L}_{concat}$ for the concatenated representation $C(\phi'(X) \oplus \alpha(D))$ and $\mathcal{L}_{nv}$ for the non-visual embedding $C(\alpha(D))$ only. We have observed that with only one loss function for the concatenated representation the model tends to mostly rely on the video

Algorithm: Difficulty Level Selection for Next Problem.

```
A ← sorted array of problem difficulties in ascending order;
DFM(video, m_k, d_k, d_{k+1}) ← deep fusion model predicting outcome;
lowerBound ← 0, upperBound ← length(A) - 1;
do
    nextDifficulty ← ( upperBound + lowerBound ) // 2;
    outcome ← DFM(video, m_k, d_k, A[nextDifficulty]);
    if outcome is positive then
        return A[nextDifficulty];
    else if outcome is negative then
        upperBound ← nextDifficulty - 1;
    end
while lowerBound < upperBound;
return A[nextDifficulty];
```

representation to make predictions while failing to fully utilize the non-visual information. This could be due to the visual features being more salient or easier to learn.

Hence we add one additional loss function for the non-visual information, which encourages the model to focus on the non-visual information and learn to incorporate their features more effectively. The overall loss function

$$\mathcal{L} = \lambda_1 \mathcal{L}_{concat} + \lambda_2 \mathcal{L}_{nv},$$

where λ_1, λ_2 are scale factors.

Implementation Details of Deep Fusion Model. We implemented our models in PyTorch and conducted experiments on one NVIDIA TITAN Xp GPU. For Facial Analysis Network, we used the official implementation[1] of OpenFace [3] to extract three-dimensional head location and rotation, three-dimensional eye gaze, and the presence and the intensity of 18 facial action units for each video. Overall, a 49-dimensional feature vector was extracted for each frame. For the Affect Network, we pre-trained a ResNet-50 [8] network on 50,000 randomly sampled images from a labeled facial expression dataset, the AffectNet dataset [14] and validated on 5,000 randomly selected images. We extracted the affect embedding from our videos by performing inference of the Affect Network on each frame. The dimensionality for both the video embedding and the non-visual embedding is set to 200. Following [19], when training the model, we downsampled the videos to three frames per second from the original 30 frames per second, to reduce both processing time and storage requirements. We trained the full model using the Adam optimizer [11] with β_1 of 0.9, and β_2 of 0.999. We used a learning rate of 3×10^{-5} for 100 epochs, and a batch size of 1 following [19].

3.2.5 Evaluation of Deep Fusion Model. To evaluate the performance of the Deep Fusion Model of COVES, we implemented three baseline methods for comparison. The majority vote classifier selects the majority class in the dataset, "Positive." The random guess classifier randomly determines the outcome according to the frequency distribution of the two classes in the dataset. The third model of non-visual data trains a classifier on top of the non-visual embedding $C(\alpha(N))$, considering non-visual data only. We also reproduce the state-of-the-art method for outcome prediction, ATL-BP [18], and report its performance on the dataset. As ATL-BP uses visual information only and has an identical network structure to the video analysis module of COVES, the result for ATL-BP also shows the performance of the visual branch of our model without multimodal fusion.

Following the experimental setup [18], we performed five-fold cross-validation by randomly shuffling the data samples and constructing five testing and training sets, where the training set contains 80% of the data and the testing set contains the rest, 20% data. The accuracy, precision, recall, mean F_1-score, and Cohen's Kappa coefficient are reported in Table 1. The COVES model consistently improves over all the baselines and previous state-of-the-art ATL-BP [18] for all the evaluation metrics.

We assessed how the difficulty level d_{k+1} of the next problem impacts the model prediction result. We changed the value of d_{k+1} while keeping the values of other variables unchanged, and observed how the model output changes. Intuitively, if a more difficult problem is given to the student, the outcome is expected to be negatively affected. Similarly, an easier problem tends to positively affect the outcome. The simulation results show a similar trend with this assumption. By increasing d_{k+1} to the highest level, the prediction results of 85.4% of positive samples change to negative. By decreasing d_{k+1} to the lowest level, the prediction results of 60.9% of negative samples change to positive. If we increase the level d_{k+1} of the negative samples, all of them remain negative; if we decrease the level d_{k+1} of the positive samples, all of them remain positive. This assessment verifies that the difficulty level d_{k+1} largely contributes to the model output, which enables the algorithm to select the difficulty level that yields a positive outcome.

3.3 Model Integration and Deployment

Finally, we integrated COVES into MathSpring. To protect student privacy, the analysis of student videos is only conducted locally, in the front end of the web-based software. No video is stored on the web. The client will first download the parameters of the COVES model from our server, and then start to analyze the real-time video and learning log data of the student locally. No video data is uploaded to the MathSpring server and only the selected problem difficulty result is returned.

To achieve real-time prediction, COVES must perform with high efficiency. We applied several optimization techniques to improve model efficiency. We first reduced the frame rate of the video from 3 FPS to 1 FPS to speed up the analysis. We also used model quantization to reduce the model size while accelerating the inference time of the model. Specifically, model quantization executes some of the operations in the model with reduced precision (8-bit integer) rather than full precision (32-bit floating point) values. By applying these two optimization techniques, the initial downloading time of the model is reduced from 40 seconds to less than 10 seconds, and the algorithm is able to achieve real-time processing. Table 2 shows the performance of COVES after FPS reduction and quantization. We can observe a slight performance drop (1% – 5%) compared to the original model, which is a reasonable trade-off between performance and efficiency.

[1] https://github.com/TadasBaltrusaitis/OpenFace

Table 1: Results for outcome prediction on the MathSpring Children Dataset using five-fold cross-validation.

Method	Mean F-Score	Accuracy	Precision	Recall	Cohen's Kappa
Majority Vote	0.365	57.5%	0.287	0.5	0
Random Guess	0.518	53.2%	0.520	0.519	0
Non-visual Data	0.587	60.3%	0.607	0.594	0.191
ATL-BP [18]	0.600	63.3%	0.623	0.603	0.216
Deep Fusion Model of COVES	**0.645**	**65.2%**	**0.647**	**0.648**	**0.294**

Table 2: Performance of the prediction model on the MathSpring Children Dataset using five-fold cross-validation after frame rate reduction and quantization.

Method	Mean F-Score	Accuracy	Precision	Recall	Cohen's Kappa
Deep Fusion Model of COVES	**0.645**	**65.2%**	**0.647**	**0.648**	**0.294**
Deep Fusion Model of COVES After FPS Reduction	0.637	65.1%	0.642	0.638	0.278
Deep Fusion Model of COVES After Quantization	0.632	64.4%	0.636	0.633	0.267

Table 3: Activity time and mastery level for students in treatment and control groups. Results of ANOVA test for time, number of attempts and mastery level for students using the COVES (treatment) and legacy EBT algorithm (control).

Dependent Variable	p-value	F	M: Control	M: Treatment
Time Solving Problems	< 0.05	17.97	120,088	62,127
Time Interacting with Tutor	< 0.05	11.13	30.20	34.87
Time until First Hint	0.08	3.16	486,214	277,006
Time until First Attempt	< 0.05	16.87	115,353	59,865
Number of Attempts	< 0.05	5.07	1.53	1.17
Level of Mastery	< 0.05	9.65	0.32	0.43
PostTest is Correct	< 0.05	29.01	4.22	5.00

4 REAL-TIME CLASSROOM EXPERIMENT

With the COVES-integrated MathSpring platform, we were able to "close the loop" for computer-vision-based tutoring systems and test the tutor's decision making in a real-time classroom experiment.

4.1 Design of Classroom Experiment

During the summer of 2022, N=22 seventh-grade students from a national summer camp program participated in the study. The camp, held at two colleges in North Eastern U.S., serves girls from low-income families and provides opportunities to engage in building skills and confidence to promote college and career readiness. The students worked with MathSpring for at least 30 minutes during one two-hour workshop. Students accessed MathSpring on a laptop and solved problems based on common core math standards according to their grade level. Before students began, one researcher provided a tutorial on the software. Each student was randomly assigned to either the "treatment group" (using COVES, N=13) or the control group (using the legacy algorithm EBT, N=9). Students took math tests before and after the 30-min tutor session, based on the content with which they engaged.

4.2 Results of Classroom Experiment

Results of the classroom experiment show the post-test scores (M = 4.682; SD = 2.276) were higher than the pre-test scores (M = 4.091; SD = 1.950) for all students. When comparing control and treatment groups, the pre-test indicated that no significant difference existed between them, suggesting that they were similar in terms of math test scores prior to the intervention. Post-test scores (M = 5.000; SD = 2.081) for students in the treatment group were significantly higher than the post-test scores (M = 4.222; SD = 2.587) for students in the control group, indicating the COVES problem selector led to better learning outcomes according to students' math test performance.

Additionally, significant differences were found in the scores between students in the treatment and control group for multiple dependent variables including time solving problems, time interacting with tutor, time until first attempt, number of attempts, and level of mastery, see Table 3. Of note, participants in the treatment group tended to spend less time per problem, while requiring fewer attempts to solve problems correctly when compared to the control group. While they spent less time per problem, participants in the treatment group spent more time interacting with the MathSpring pedagogical agent , a learning companion to support students' affect and learning process. Correlation analysis also indicated a positive correlation between the amount of time spent with the agent and the level of mastery in the treatment group ($R = 0.28; p < 0.05$). Additionally, participants in the treatment group needed fewer attempts to solve problems than those in the control group.

The difference in time spent per problem, the time until the student's first attempt, and the variance in the total number of attempts before a solution was achieved suggest that students in the treatment group, who used COVES, received problems better adapted to their ability than did students in the control group. We propose that since students in the treatment group received problems at an appropriate difficulty level, the tutor (more knowledgeable other) kept students within the zone of proximal development (ZPD), leading

to more student engagement and "flow" (i.e., total concentration and absorption in the task at hand [6]).

Conversely, we propose that students in the control group may have received problems that were too difficult, pushing them outside of their zone, leading to frustration and a lack of desire to continue the interaction. Additional evidence that students in the treatment group received more appropriate problems can be seen in the difference in the mastery level; students in the treatment group had a significantly higher mastery level than did students in the control group (mean difference of 0.11 with a p value of <0.05). These results suggest that students in the treatment group felt more comfortable, received problems that better kept them in the ZPD, and overall had a more successful experience with the system.

5 DISCUSSION AND CONCLUSIONS

We developed and evaluated COVES, a deep learning model that predicts a student's ability to solve math problems and personalizes content difficulty for the student in real time by analyzing a combination of visual information and log data. Specifically, we propose a novel deep fusion model that combines visual affective analysis and student log data with a deep attention-based fusion network The deep fusion model predicts students' engagement with problems as either productive (solving problems correctly, after hints, or after a single incorrect attempt) or unproductive (quick-guessing, giving up, skipping the problem or rushing to answer). The problem difficulty that yields a positive engagement outcome is selected based on prediction results given a student's previous behavior and the difficulty of the candidate next problem.

The technical challenge in designing the deep fusion model was for us to find a way to balance the interpretation of the visual and non-visual information, so that both can effectively contribute to the outcome prediction. The dimensionality of the video input is significantly larger than that of the log data, so it had to be significantly reduced. Furthermore, we created an attention module to generate an attention-boosted video representation that we then concatenated with a log data embedding of equal size. The overall loss function used to train the deep fusion model is a combination of two weighted cross-entropy losses, one for the fusion module, the other for the non-visual embedding network.

We trained our deep fusion model on the MathSpring Children Dataset consisting of 968 videos of fifty-one sixth-grade students as well as their problem solving outcomes and log data. Evaluation results indicate that the proposed deep fusion model achieves superior performance compared with baselines on outcome prediction. We then integrated the COVES model back into the frontend of the MathSpring tutoring system, by applying several optimization techniques to improve model efficiency for real-time prediction.

By integrating COVES into the MathSpring platform, we were able to "close the loop" for computer-vision-based tutoring systems and demonstrate the potential for computer vision to be used not just for post-experiment analysis, but as an integral part of the real-time decision making by the tutor. To achieve real-time performance of the tutor, we had to overcome the technical challenge of processing video frames to compute attention-boosted visual embeddings in real time. We were able to do that by reducing the video frame rate and quantizing the model to integer precision.

We evaluated the effectiveness of the COVES-enhanced tutor in a real-time classroom experiment. According to pedagogy research [22], learners perform best when tasks are just beyond their capabilities or out of their ability range, if they are helped by a "more knowledgeable other." The tutor embedded with COVES served as such a knowledgeable other. The students who used the tutor embedded with COVES received problems at an appropriate difficulty level and were kept within a zone of proximal development, leading to more student engagement, flow, and potential cognitive growth. Conversely, students in the control group, who used the tutor without COVES, may have received problems that were too difficult, pushing them outside of their zone, leading to frustration and a lack of desire to continue the interaction. Results of the classroom experiment show that students had a higher mastery level when the COVES model had access to their faces and gestures, suggesting that COVES-selected problems were better adapted to students' abilities than problems selected without COVES. Quantitative analyses indicates that COVES led to higher levels of engagement, faster problem solving, less time spent before a student's first attempt at a solution, less time before a student asked for a first hint, and fewer attempts before a student chose the correct solution. These results suggest that students in the COVES group received more appropriate problems better adapted to their ability. This result combined with the statistically significant difference in time spent interacting with the tutor suggests that students in the treatment group felt more comfortable, received problems that better maintained them in the zone of proximal development, and overall had more successful experiences with the tutor.

ETHICAL IMPACT STATEMENT

Ethical dimensions exist in the design, development, and deployment of AI systems for education. Bias can permeate data collection, data analysis and usage. Computer vision systems are highly dependent on training datasets that might learn and amplify biases, including prejudices against individuals or group defined on protected attributes (gender, ethnicity, race, sexual orientation) [7, 16]. We considered the research team's own blind spots, since teams often struggle to anticipate the sub-populations they might inadvertently miss. The girls and boys whose images are included in the dataset used to train COVES (MathSpring Children Dataset [18]) are members of various ethnic and racial groups (African American, Hispanic, Caucasian). The processing of videos of students was done with the consent of students, parents and other stakeholders (teachers, administrators). Many students readily offer much more sensitive personal data to social media applications.

Increasingly widespread online instructional systems present many challenges and have the potential to amplify social inequities or even create new ones. It is difficult to collect enough data to represent different contexts, cultures, and countries; different demographic groups might have different ways of responding to online activities. We remain vigilant in selecting demographic groups to consider and in building multi-disciplinary awareness of ethics.

ACKNOWLEDGMENTS

We acknowledge funding for this work by the U.S. National Science Foundation, grants # 1551572, # 1551590, # 1551589, and # 1551594.

REFERENCES

[1] Ivon Arroyo, Hasmik Mehranian, and Beverly P Woolf. 2010. Effort-based tutoring: An empirical approach to intelligent tutoring. In *3rd International Conference on Educational Data Mining Pittsburgh, PA, USA June 11-1-3.*

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1409.0473

[3] Tadas Baltrusaitis, Amir Zadeh, Yao Chong Lim, and Louis-Philippe Morency. 2018. OpenFace 2.0: Facial Behavior Analysis Toolkit. In *13th IEEE International Conference on Automatic Face & Gesture Recognition*. 59–66. https://doi.org/10.1109/FG.2018.00019

[4] Laura E Berk and Adam Winsler. 1995. Vygotsky: His life and works and Vygotsky's approach to development. *Scaffolding children's learning: Vygotsky and early childhood learning* (1995), 25–34.

[5] Byung-In Cho, Joel A Michael, Allen A Rovick, and Martha W Evens. 1999. A Curriculum Planning Model for an Intelligent Tutoring System.. In *The Florida AI Research Society (FLAIRS-12) Conference, Orlando, FL*. 197–201.

[6] Mihaly Csikszentmihalyi. 1990. *Flow: The psychology of optimal experience*. Harper & Row New York.

[7] Simone Fabbrizzi, Symeon Papadopoulos, Eirini Ntoutsi, and Ioannis Kompatsiaris. 2022. A survey on bias in visual datasets. *Computer Vision and Image Understanding* 223 (2022), 1–16. https://doi.org/10.1016/j.cviu.2022.103552.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] Ajjen Joshi, Danielle Allessio, John Magee, Jacob Whitehill, Ivon Arroyo, Beverly Woolf, Stan Sclaroff, and Margrit Betke. 2019. Affect-driven Learning Outcomes Prediction in Intelligent Tutoring Systems. In *The 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), Lille, France, May*. IEEE, 1–5.

[11] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[12] Andrew S Lan and Richard G Baraniuk. 2016. A Contextual Bandits Framework for Personalized Learning Action Selection. In *The 9th International Conference on Educational Datamining (EDM), Raleigh, NC, USA, June 29–July 2*. 424–429.

[13] Saul McLeod. 2020. *Vygotsky's Sociocultural Theory of Cognitive Development*. Simply Psychology. https://www.simplypsychology.org/vygotsky.html.

[14] Ali Mollahosseini, Behzad Hassani, and Mohammad H. Mahoor. 2019. AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. *IEEE Transactions on Affective Computing* 10 (2019), 18–31.

[15] Tom Murray and Ivon Arroyo. 2002. Toward measuring and maintaining the zone of proximal development in adaptive instructional systems.. In *International Conference on Intelligent Tutoring Systems*. Springer, 749–758.

[16] Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasanakis, Ioannis Kompatsiaris, Katharina Kinder-Kurlanda, Claudia Wagner, Fariba Karimi, Miriam Fernandez, Harith Alani, Bettina Berendt, Tina Kruegel, Christian Heinze, Klaus Broelemann, Gjergji Kasneci, Thanassis Tiropanis, and Steffen Staab. 2020. Bias in data-driven artificial intelligence systems—An introductory survey. *WIREs Data Mining and Knowledge Discovery* 10, 3 (2020), e1356. https://doi.org/10.1002/widm.1356

[17] Matteo Orsoni, Alexander Pögelt, Nghia Duong-Trung, Mariagrazia Benassi, Milos Kravcik, and Martin Grüttmüller. 2023. Recommending Mathematical Tasks Based on Reinforcement Learning and Item Response Theory. In *International Conference on Intelligent Tutoring Systems*. Springer, 16–28.

[18] Nataniel Ruiz, Hao Yu, Danielle A Allessio, Mona Jalal, Ajjen Joshi, Tom Murray, John J Magee, Kevin Manuel Delgado, Vitaly Ablavsky, Stan Sclaroff, Ivon Arroyo, Beverly P Woolf, Sarah Adel Bargal, and Margrit Betke. 2023. ATL-BP: A Student Engagement Dataset and Model for Affect Transfer Learning for Behavior Prediction. *IEEE Transactions on Biometrics, Behavior, and Identity Science* 5, 3 (2023), 411–424. DOI 10.1109/tbiom.2022.3210479.

[19] Nataniel Ruiz, Hao Yu, Danielle A Allessio, Mona Jalal, Ajjen Joshi, Thomas Murray, John J Magee, Jacob R Whitehill, Vitaly Ablavsky, Ivon Arroyo, Beverly P Woolf, Stan Sclaroff, and Margrit Betke. 2021. Leveraging Affect Transfer Learning for Behavior Prediction in an Intelligent Tutoring System. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021) December 15–18*. IEEE, 1–8. DOI 10.1109/FG52635.2021.9667001.

[20] Sandra Sampayo-Vargas, Chris J Cope, Zhen He, and Graeme J Byrne. 2013. The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game. *Computers & Education* 69 (2013), 452–462.

[21] Avi Segal, Yossi Ben David, Joseph Jay Williams, Kobi Gal, and Yaar Shalom. 2018. Combining difficulty ranking with multi-armed bandits to sequence educational content. In *International Conference on Artificial Intelligence in Education*. Springer, 317–321.

[22] Lev Semenovich Vygotsky. 1987. *Thinking and speech. In R.W. Rieber & A.S. Carton (Eds.), The collected works of L.S. Vygotsky, Volume 1: Problems of general psychology. pp. 39–285.* New York: Plenum Press.

[23] Yaqian Zhang and Wooi-Boon Goh. 2019. Bootstrapped policy gradient for difficulty adaptation in intelligent tutoring systems. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Montreal, QC, Canada*. 711–719.

[24] Yaqian Zhang and Wooi-Boon Goh. 2021. Personalized task difficulty adaptation based on reinforcement learning. *User Modeling and User-Adapted Interaction* 31, 4 (2021), 753–784.