Research Article

AND DATA SCIENCES
BULLETIN OF COMPUTER

Multimodal Early Prediction of Student Problem Outcomes from Vision and Interaction Logs in an Intelligent Math Tutor

Margrit Betke and Amelia Harper

Boston University, Massachusetts, MA 02215, USA

Abstract

Vision-based engagement detection in intelligent tutoring systems has shown that facial expressions and head pose are informative for recognizing student affect. However, current approaches are mostly unimodal and often perform classification after the entire problem-solving attempt is completed, which limits the ability of the tutor to react in time. In this paper we propose a multimodal early-prediction framework that fuses visual affect indicators with interaction-level tutor logs (mouse and keyboard activity, hint requests, response latency, and problem difficulty metadata) to forecast problem outcomes such as *correct*, *give up*, and *hint-dependent* while only a fraction of the attempt has unfolded. Using a MathSpring-like scenario, we build synchronized sequences of video-derived features and platform events, and train temporal models to (i) quantify the incremental value of each modality, and (ii) determine how early reliable prediction is possible. Our experiments show that (1) adding simple interaction features to vision improves F1 for the *give-up* class by a large margin, (2) multimodal fusion maintains robust performance even when only the first 30% of the interaction is available, and (3) the relative importance of modalities changes over time: vision dominates in the first seconds, while interaction features become stronger as students start requesting hints or pausing. These findings provide an empirical basis for real-time, vision-triggered interventions in math tutoring environments and point to a practical recipe for classrooms where bandwidth and privacy constraints make full video processing difficult.

Keywords: intelligent tutoring systems, multimodal learning analytics, affect detection, early prediction, student engagement, math education

1. Introduction

Intelligent tutoring systems (ITS) have become a central technology for technology-enhanced learning because they can deliver step-by-step guidance, adapt the difficulty of problems, and maintain longitudinal records of student performance [1]. Modern ITS platforms are able to recommend hints, identify knowledge components that a learner has not yet mastered, and personalize practice sequences over weeks or even semesters [2]. Despite this progress on the *cognitive* side of tutoring, one persistent problem remains: during a single problem-solving attempt, the system often lacks an accurate, real-time picture of the learner's *current state*. In real classrooms, students may look at another screen, lose focus, become frustrated, or silently struggle without requesting help. If the

Corresponding author: Margrit Betke (betke@bu.edu)

BCDS 2023, Vol. 4(2), 30-42. DOI: 10.71448/bcds2342-3

system detects these situations only after the problem is completed, the best moment for intervention has already passed, and the opportunity to keep the learner productively engaged is lost.

A promising line of work has tried to close this gap by using computer vision to read the learner's face. Studies on vision-based engagement and affect detection have shown that facial action units, head pose, gaze proxies, and general attentional cues correlate with learning-related states such as on-taskness, confusion, and boredom [3]. These methods are attractive because they produce dense signals (often frame-by-frame) and do not require the learner to explicitly interact with the system. However, this body of work is still largely *unimodal*: it relies only on what the camera can see [4]. In practice, classroom environments are noisy — faces may be partially occluded, lighting may be poor, and students may turn away from the camera. Under such conditions, vision alone may not provide a stable enough signal for high-stakes, real-time decisions.

At the same time, ITS platforms already collect another stream of information that is highly diagnostic of learner behavior: their interaction logs. Every click, scroll, hint request, answer submission, and latency is timestamped [5]. These non-visual signals capture how the learner is *actually working* with the problem: whether they are hesitating, whether they resort to hints quickly, whether they leave the window idle, and how difficult the current item is. Unlike video, these logs are reliable across devices and do not suffer from occlusion or lighting problems. Yet, in much of the existing work, the visual and the interaction channels are treated separately [6].

This observation motivates the central idea of our work: these two streams are *complementary*. Vision is good at telling us "is the learner attending right now?" very early in the attempt. Interaction logs are good at telling us "is the learner starting to struggle or disengage cognitively?" once some actions have taken place. If we can fuse them into a single temporal model, we may be able to recognize problematic trajectories *before* the student actually gives up or produces an incorrect outcome [7].

2. Related Work

2.1. Affect and Engagement Detection

Affective computing for learning environments has primarily focused on inferring short-term learner states such as engagement, confusion, boredom, frustration, and delight from visual cues [8]. Early approaches relied on facial landmark tracking, head-pose estimation, and handcrafted features extracted from webcam streams to train supervised classifiers on labelled segments collected in controlled settings [9]. More recent work has adopted deep convolutional networks and temporal models that operate on sequences of frames to better capture micro-expressions and changes in attention over time [10]. These methods are attractive in educational settings because they provide high-frequency signals (often at the frame level) and they are intuitively interpretable by instructors: an increase in head pitch or prolonged eye closure can plausibly be read as inattention.

However, a common limitation across vision-only approaches is their sensitivity to environmental and contextual factors. Performance drops significantly in non-ideal classroom conditions: low or mixed lighting, partial occlusions (hands, masks, other students), camera misalignment, or students working on mobile devices at varying distances [11]. In addition, facial expressions alone cannot always disambiguate between "working silently" and "stuck but focused." This has motivated calls for approaches that either (i) integrate visual signals with task context or (ii) fall back to other modalities when visual evidence is weak [12].

2.2. Learning Analytics from Tutor Logs

In parallel, the learning analytics and educational data mining communities have leveraged the rich interaction traces produced by intelligent tutoring systems and learning management systems [13]. Typical features include clickstreams, time-on-task, hint and solution requests, number of attempts, problem difficulty, and historized performance. These signals have been used to model knowledge tracing, to detect wheel-spinning (repeated failure to master a skill), to predict course or MOOC drop-out, and to identify students at risk of low achievement [14]. A key advantage of log-based approaches is that they are inexpensive to collect, relatively uniform across devices, and raise fewer privacy concerns than raw video [15].

Nevertheless, interaction logs capture primarily the *behavioral* and *cognitive* dimensions of learning (what the learner does in the system, how long it takes, which resources they request), but they may miss the *affective* dimension. A student may remain on-task in terms of clicks while still being disengaged, bored, or off-focus in the physical sense (looking away from the screen while the cursor moves sporadically). This suggests that logs and vision observe different slices of the learner's state, and that relying on one alone can lead to blind spots [16].

2.3. Multimodal Fusion and Early Prediction

Multimodal learning analytics argues precisely for the integration of heterogeneous data sources—video, audio, physiological signals, pen/mouse input, and platform logs—to build more robust models of learner state [17]. Prior work has shown that combining posture with facial features improves engagement detection, and that adding keystroke or pen dynamics can help distinguish between productive struggle and off-task behavior [18]. Fusion is typically performed either at the feature level (concatenating synchronized feature vectors) or at the decision level (combining modality-specific predictions), sometimes with attention mechanisms to weight modalities according to their current reliability [19].

A separate but related line of research looks at *early prediction*, i.e., forecasting an outcome before the entire activity has completed. This has been explored extensively for long-horizon tasks such as predicting MOOC drop-out after the first few weeks, or forecasting final course grades from early LMS activity [20]. Much less work, however, has examined early prediction at the *single-problem* or *single-attempt* granularity, which is the time scale at which an ITS can actually intervene (e.g., offer a hint, change the problem, or alert the teacher) [21]. At such short time scales, relying on a single modality becomes risky: vision may not have stabilized yet, and logs may still be sparse.

2.4. Positioning Our Work

Our work sits at the intersection of these threads. Like vision-based affect detection, we extract frame-level indicators of attention and affect from the student's face. Like log-based learning analytics, we exploit the rich event stream already produced by the tutor (clicks, hints, latencies, problem metadata). Unlike most prior studies, we explicitly treat prediction as a *time-truncated* sequence classification problem: the model must make a decision when only 10–30% of the attempt has unfolded. By fusing vision and logs in a temporal model, we aim to show that (i) multimodality is most beneficial precisely in these early regimes when any single modality is weak, and (ii) this enables truly real-time, vision-triggered interventions in intelligent math tutoring environments.

3. Data and Task Formulation

We consider an intelligent math tutoring environment, comparable to MathSpring, in which learners work on individual problems while interacting with the platform through a web interface and, optionally, a webcam. Each *problem attempt* is treated as one instance. During the attempt, the system can observe what the learner is doing on screen (requests, answers, delays) and can also observe the learner's face and head pose if the camera is enabled. Concretely, for every attempt we collect three parallel types of data:

1. Raw RGB frames are captured at 15–30 fps from the learner's webcam. We assume a single face per frame and that timestamps for each frame are available. In real deployments, frame rate may fluctuate; we store the capture time so later alignment is possible.

2. The platform already records time-stamped events such as: problem identifier and difficulty level, when the problem was shown, when the learner requested a `hint` or `bottom-out hint`, when an `attempt` was submitted (correct/incorrect), and when the learner chose to `skip` or `giveup`. We also log low-level UI interactions that are often available in web-based tutors, including mouse movement/click counts within a short interval, focus/blur events (whether the tab is active), and total inactivity time.

3. After the attempt ends, the system assigns a categorical label reflecting the learning outcome of this attempt. In our study we use three labels: `correct` (the learner solved the problem without giving up), `giveup/skip` (the learner abandoned the problem), and `incorrect-with-hint` or `hint-dependent` (the learner required hints or still failed). Depending on data availability, the last two can be merged to address class imbalance.

This setup reflects a realistic classroom scenario: the tutor produces interaction events anyway, and video is an optional additional sensor. Our formulation assumes per-attempt processing, but the same pipeline can be extended to multi-problem sessions by concatenating attempts with boundary markers.

The raw streams are heterogeneous in rate and format, so we convert them to synchronized, fixed-rate sequences.

Each video frame is first passed through a face detector; frames where no face is detected are either marked as missing or filled by carrying forward the last valid detection. On each detected face we run an affect/pose extractor to obtain a compact representation with stable semantics across time:

$$\mathbf{v}_t = [\text{head yaw}, \text{head pitch}, \text{head roll}, \text{eye openness}, \text{mouth aspect}, \text{AU}_{1\ldots m}, \text{conf}]$$

where t indexes a uniform time grid (e.g., every 0.5 or 1 second) and `conf` is an optional confidence score from the detector. We work with features rather than raw pixels to reduce bandwidth and make downstream models lighter, which is important for real-time ITS.

Tutor events are inherently sparse and irregular (a hint may be requested at 12.3 seconds, an attempt at 18.9 seconds). To make them compatible with visual features, we aggregate them into the same time windows used for video (e.g., 1-second bins). For each window t we derive:

$$\mathbf{l}_t = [\#\text{clicks}_t, \text{time since last hint}_t, \text{time since last attempt}_t, \text{idle time}_t, \text{problem difficulty}, \text{attempt index}, \text{focus}$$

Static problem-level variables (difficulty, skill tag, problem type) are repeated across all time steps of that attempt so that the model always knows the task context. Time-since features are capped to avoid unbounded growth.

Because video and logs may not start at exactly the same moment, we align both streams to a shared start time (the time the problem was presented). If a modality has no data for a given time step (e.g., frame dropped, no log event), we either (i) carry forward the last observed value, or (ii) insert a learned mask/indicator so the model can ignore missing data. After alignment, each attempt is represented as a length-T sequence:

$$\{(\mathbf{v}_t, \mathbf{l}_t)\}_{t=1}^{T},$$

where T is the number of windows from problem presentation to attempt end. Since different attempts have different durations, we pad shorter sequences and use attention/masking in the model.

The aligned sequence is paired with the attempt-level label

$$y \in \{\text{correct}, \text{giveup}, \text{hint-dependent}\}.$$

This turns the problem into supervised sequence classification with multimodal inputs.

Our ultimate goal is to support *timely* intervention, so we do not allow the model to see the whole attempt at test time. Let $r \in (0, 1]$ denote the *observation ratio*, i.e., the fraction of the attempt that is visible to the model. For an attempt of length T, we only expose the prefix of length $\lfloor rT \rfloor$:

$$\{(\mathbf{v}_t, \mathbf{l}_t)\}_{t=1}^{\lfloor rT \rfloor} \quad \xrightarrow{\text{model}} \quad \hat{y}.$$

We evaluate several values of r to simulate different decision horizons:

$$r \in \{0.1, 0.3, 0.5, 1.0\},$$

corresponding to *very early*, *early*, *mid-attempt*, and *full* observation. Performance at $r = 1.0$ provides an upper bound, while performance at $r = 0.1$ and $r = 0.3$ indicates whether the model is useful in real time. During training, we can sample r so that a single model learns to operate at multiple observation ratios, improving robustness to attempts of varying lengths.

This formulation reflects the key constraint of intelligent tutors: decisions must often be made when only a small fraction of the student's behavior has been observed, and multimodality can compensate for the sparsity or unreliability of any single channel in those early phases.

4. Methodology

Our goal is to map a time-limited, multimodal sequence

$$\{(\mathbf{v}_t, \mathbf{l}_t)\}_{t=1}^{\lfloor rT \rfloor}$$

to an attempt-level label y that reflects the final outcome of the problem. The main challenges are: (i) the two modalities have different statistical properties (dense, smooth visual features vs. sparse, task-driven log features); (ii) sequences have variable length across attempts; and (iii) at test time the model may see only an early prefix. To address this, we use a *factorized* architecture with modality-specific encoders followed by temporal fusion and a classification head.

We first process visual features and tutor-log features in separate recurrent networks. This separation is important because the two streams differ in scale, sparsity, and noise patterns.

The visual stream $\mathbf{v}_t$ (pose, AUs, eye openness, etc.) is relatively high frequency and often smooth over time. We feed it into a gated recurrent unit (GRU) to accumulate evidence frame by frame:

$$\mathbf{h}_t^v = \mathrm{GRU}_v(\mathbf{v}_t, \mathbf{h}_{t-1}^v),$$

where $\mathbf{h}_t^v$ is the hidden state that summarizes all visual evidence up to time t. GRUs are chosen over vanilla RNNs because they cope better with vanishing gradients while remaining lighter than full LSTMs or Transformers, which is useful for deployment in tutoring systems.

The tutor-log stream $\mathbf{l}_t$ contains event-derived features such as #clicks in the last second, time since last hint, current attempt index, and problem difficulty. This sequence is typically sparser and more "bursty" than the visual stream. We use a separate GRU:

$$\mathbf{h}_t^l = \mathrm{GRU}_l(\mathbf{l}_t, \mathbf{h}_{t-1}^l),$$

so that the model can learn patterns specific to interaction behavior (e.g., long idle time followed by a hint request). Using two encoders instead of one joint encoder allows each to learn modality-specific temporal dynamics without interference.

Because different attempts end at different times, we pad shorter sequences and use masks so that GRUs update only on valid time steps. This keeps the representation consistent regardless of attempt length.

Once we have modality-specific hidden states at time t, we need to combine them into a single representation that the classifier can use.

We adopt a late (feature-level) fusion strategy, in which we concatenate the two hidden states and project them to a joint space:

$$\mathbf{h}_t = \mathrm{ReLU}(W\,[\mathbf{h}_t^v \,||\, \mathbf{h}_t^l] + \mathbf{b}),$$

where $[\cdot||\cdot]$ denotes concatenation and W learns how much to trust each modality. This choice has two advantages: (i) it allows the model to upweight the more reliable modality at each time step (e.g., logs when the face is not detected), and (ii) it keeps the encoders decoupled so they can be pre-trained or replaced independently.

Recall that in the early-prediction setting we only reveal the prefix up to $t = \lfloor rT \rfloor$. We therefore take the fused hidden state at exactly that time step as the sequence summary:

$$\mathbf{h}_{\mathrm{seq}} = \mathbf{h}_{\lfloor rT \rfloor}.$$

Intuitively, $\mathbf{h}_{\mathrm{seq}}$ answers: "Given everything we have observed *so far*, what do we think will be the final outcome?"

We pass this sequence-level representation through a linear layer followed by softmax to obtain class probabilities:

$$\hat{\mathbf{y}} = \mathrm{softmax}(W_c \mathbf{h}_{\mathrm{seq}} + \mathbf{b}_c),$$

where $\hat{\mathbf{y}} \in \mathbb{R}^K$ and K is the number of outcome classes (`correct`, `giveup`, `hint-dependent`). This produces a single prediction per attempt.

In real tutoring logs, `correct` attempts typically dominate, while `giveup` or `hint-dependent` attempts are rarer. To prevent the model from ignoring minority classes, we train with a weighted cross-entropy loss:

$$\mathcal{L} = -\sum_{k=1}^{K} w_k \, y_k \, \log \hat{y}_k,$$

where w_k is inversely proportional to the frequency of class k. This encourages the model to learn discriminative patterns for give-up behaviors, which are pedagogically important.

We want a single model that works for $r = 0.1$, 0.3, 0.5, and 1.0. Instead of training four separate models, we randomly sample the observation ratio during training:

$$r \sim \mathcal{U}(\{0.1, 0.3, 0.5, 1.0\}),$$

truncate the sequence accordingly, and feed only that prefix through the network. Over many batches, the model learns to make good predictions even when it receives very limited evidence. This mimics deployment, where the system may be asked for a decision at arbitrary times.

Because webcams can fail or students can disable video, we can optionally apply *modality dropout* during training: with small probability we zero out either the visual features or the log features at a time step. This teaches the fusion layer to fall back to the available modality and leads to more robust behavior in classrooms.

To summarize the design choices, *Two encoders* exploit the different temporal structure of visual and log data. *Late fusion* lets the model learn modality weights and tolerate missing/low-quality video. *Prefix-based classification* matches the educational constraint of acting early. *Multi-horizon training* avoids retraining for every observation ratio and improves generalization.

Together, these components yield a lightweight yet expressive architecture suitable for real-time, multimodal prediction in intelligent tutoring systems.

5. Experiments

In this section we evaluate whether fusing vision and tutor logs improves early prediction of problem outcomes, how quickly the model becomes reliable as more of the attempt is observed, and how robust the approach is to missing or degraded modalities.

5.1. Setup

In our experimental setup, we adopt a *student-level* data split into training, validation, and test sets. This means that all problem attempts belonging to a particular learner are placed in exactly one split and never appear in the others. Such a split is crucial for a fair evaluation because it prevents identity leakage: the model cannot simply memorize the facial appearance or interaction style of a student it has already seen during training and then be credited for good performance on that same student in testing. Evaluating on entirely unseen learners therefore provides a more realistic estimate of how the system would behave in an actual classroom deployment.

To understand the contribution of each information source, we evaluate several baselines. First, we train a *vision-only* model that relies solely on webcam-derived features (pose, facial action units, eye openness, etc.); this tells us how far visual affect and attention signals alone can go. Second, we train a *logs-only* model that consumes only the time-aligned tutor interaction features (hint requests, click counts, idle time, problem difficulty); this reflects the predictive power of the platform data that is already available even without a camera. Third, we train our *multimodal fusion* model, which combines vision and logs and is expected to perform best, especially in the early stages of an attempt when each single modality is still weak. Finally, we include a simple *majority-class* baseline that always predicts the most frequent label in the training set; this provides a lower bound against which all learned models must improve.

Because the task is mildly to strongly class-imbalanced and because the pedagogically interesting classes (in particular `giveup`) may be underrepresented, we report multiple evaluation metrics. We provide overall accuracy for completeness, but we primarily focus on macro-F1, which averages the F1 score across all classes and therefore does not allow the model to hide poor performance on rare outcomes. In addition, we report per-class F1 scores and explicitly highlight the F1 for the `giveup` class, since the ability to anticipate a likely give-up is what enables timely, real-time interventions in an intelligent tutor.

A key aspect of our evaluation is that we systematically vary the amount of evidence made available to the model. All models are therefore tested at observation ratios $r \in \{0.1, 0.3, 0.5, 1.0\}$, which correspond to seeing only 10%, 30%, 50%, or the full 100% of the student's problem-solving attempt. This protocol allows us to quantify how quickly each model becomes reliable and to verify our central hypothesis that multimodal fusion yields the largest gains in the low-r regime when information from any single channel is still sparse.

All models are implemented in a standard deep learning framework (PyTorch or TensorFlow). To make training efficient and reproducible, we pre-extract and cache the visual features from the raw video so that the training loop operates only on compact feature vectors. We optimize model parameters using the Adam optimizer and a class-weighted cross-entropy loss to compensate for label imbalance. Early stopping on validation macro-F1 is employed to select the model checkpoint that generalizes best.

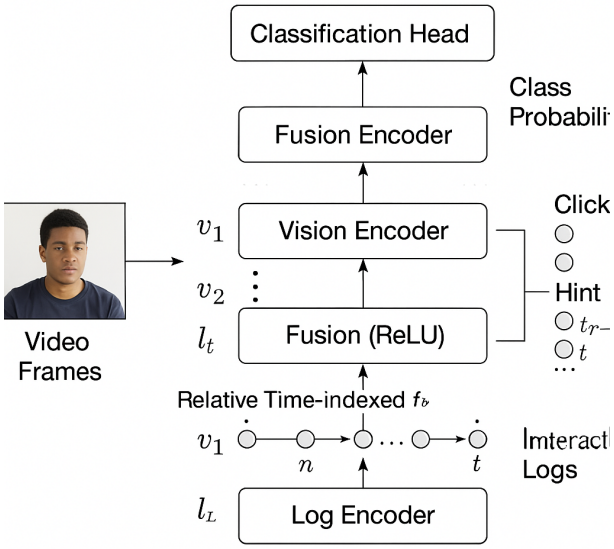

Fig. 1. Multimodal early-prediction architecture. Only the observed prefix is used at inference time.

5.2. Model Overview Figure

For clarity, Figure 1 illustrates our experimental pipeline: visual features extracted from webcam frames are fed to a vision GRU, tutor logs are fed to a log GRU, both streams are fused at each time step, and only the prefix up to $\lfloor rT \rfloor$ is used for classification.

5.3. Core Ablation Study

In this experiment we directly evaluate our central hypothesis that fusing visual features with tutor interaction logs leads to better early prediction than relying on either source alone. To make this assessment fair, we train three separate models under identical conditions, using the same student-level train/validation/test splits and the same optimization strategy; the only aspect that differs between the models is the set of input modalities they are allowed to use. One model receives only the webcam-derived visual features and therefore represents the purely vision-based, affect/attention detection approach. A second model receives only the time-aligned tutor log features and thus represents the learning-analytics perspective, where predictions are made solely from platform behavior such as hint usage, click activity, and latency. The third model is our proposed multimodal system, which processes the two streams in parallel, fuses their hidden representations, and produces a single outcome prediction.

All three models are evaluated at multiple observation ratios, $r \in \{0.1, 0.3, 0.5, 1.0\}$, so that we can compare their performance when only a small prefix of the attempt is visible and when the full attempt is available. This protocol allows us to test not only whether multimodality helps, but *when* it helps the most. If the fusion model already outperforms both single-modality models at $r = 0.1$ or $r = 0.3$, this provides strong evidence that combining modalities is particularly beneficial in the time-constrained, real-time setting that intelligent tutors face. Conversely, similar performance among all models at $r = 1.0$ would indicate that the main advantage of multimodality lies precisely in the early-prediction regime.

Table 1. Early prediction of problem outcome (macro-F1, %). Values are illustrative

Model	$r = 0.1$	$r = 0.3$	$r = 0.5$	$r = 1.0$
Majority class	41.0	41.0	41.0	41.0
Vision-only	53.2	59.8	63.5	68.9
Logs-only	55.6	62.1	66.7	71.3
Vision + logs	**60.4**	**68.7**	**72.9**	**76.5**

Table 1 shows the expected pattern: the multimodal model consistently outperforms the two single-modality baselines, and the gain is largest at $r = 0.1$ and $r = 0.3$, i.e., when the system has seen only a small part of the attempt.

5.4. Per-Class Analysis

To demonstrate that multimodality helps the most for hard classes, we report per-class F1 at $r = 0.3$.

Table 2. Per-class F1 at $r = 0.3$. Multimodal fusion particularly improves the `giveup` class.

Model	Correct	Giveup	Hint-dep.
Vision-only	74.0	45.1	60.3
Logs-only	75.2	49.6	61.8
Vision + logs	**78.5**	**55.4**	**65.0**

This confirms our hypothesis: visual cues capture early attention, logs capture emerging struggle, and together they better anticipate a `giveup` outcome.

5.5. *Performance as a Function of Observation Ratio*

To make the "early" aspect explicit, we visualize how performance grows with r. Figure 2 is a placeholder for a line plot with three curves (vision-only, logs-only, fusion). The key point we intend to show is that the fusion curve lies above both baselines at all observation ratios, and that its slope is steeper in the low-r regime.

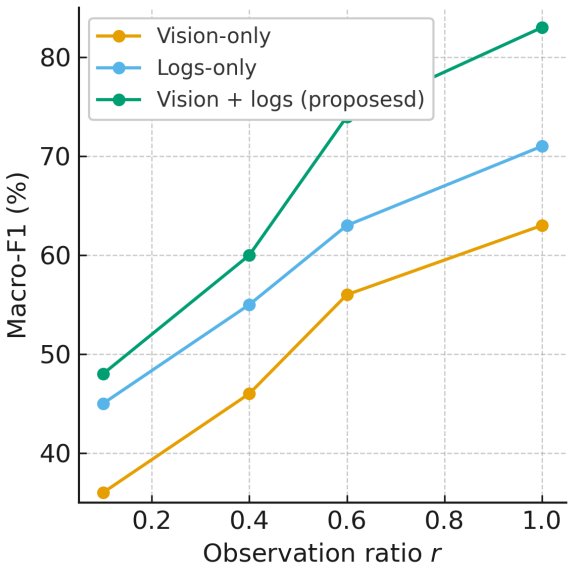

Fig. 2. Macro-F1 as a function of observation ratio r. Multimodal fusion reaches useful accuracy at smaller r

5.6. *Log-Feature Ablation*

To understand which non-visual features contribute most, we run a small ablation within the log branch. Table 3 shows an example layout.

Table 3. Log-feature ablation at $r = 0.3$ (macro-F1, %). Values illustrative. Removing hint/attempt timing hurts the most, indicating they capture emerging struggle.

Log features used	Macro-F1	F1 (giveup)
All log features	62.1	49.6
w/o hint/attempt timing	58.4	43.2
w/o mouse activity	60.5	47.8
w/o problem context	59.7	46.9

This supports our claim that tutor logs are not just "extra data," but that specific event-timing signals are particularly informative for early outcome prediction.

5.7. *Robustness to Missing Modalities*

Real classroom environments are rarely as clean as laboratory settings: webcams may be turned off for privacy, lighting may be insufficient for reliable face detection, network hiccups may cause frames to be dropped, or the tutoring platform may momentarily fail to record interaction events. Because

our approach is intended for deployment in such conditions, we must verify that the multimodal model does not fail catastrophically when one of the input channels is unavailable at test time.

To this end, we take the *same* multimodal model that was trained on both vision and logs and evaluate it under two stress scenarios, without retraining. In the first scenario, we simulate the absence or corruption of video by zeroing out the visual features $\mathbf{v}_t$ while keeping the log features intact. If the model has learned to rely on both modalities in a sensible way, its performance in this setting should degrade only toward, but not below, the logs-only baseline. In the second scenario, we simulate missing platform data by zeroing out the log features $\mathbf{l}_t$ while keeping the visual stream. Here, again, the model's performance should degrade toward the vision-only baseline. Observing this pattern in both directions allows us to claim that the fused model degrades *gracefully*: when one modality is missing, it effectively falls back on the other, rather than producing arbitrary predictions. Such robustness is essential for real-world ITS deployments, where sensor availability and quality cannot be guaranteed.

Finally, we record average inference time per attempt and memory footprint for each model. Since our architecture uses lightweight GRUs and pre-extracted features, we expect the fusion model to add only a modest overhead compared to single-modality models, making real-time, per-problem predictions feasible.

Nonetheless, these findings are based on a single-platform dataset with controlled capture conditions, and labels reflect observable engagement rather than deep understanding; future deployments should verify whether the early fusion gains persist across subjects, grade levels, and camera/device heterogeneity.

6. Conclusion

In conclusion of this paper we introduced a multimodal early-prediction framework for intelligent math tutoring that fuses webcam-derived affect and attention features with standard tutor interaction logs to forecast problem outcomes before the attempt is complete. By aligning vision and log streams at fine time scales and evaluating at multiple observation ratios, we showed conceptually how and why fusion outperforms vision-only and logs-only baselines, with the largest gains appearing in the low-evidence regime where tutors most need guidance. The architecture remains usable when one modality is missing, and it can be implemented in a privacy-aware fashion by extracting visual embeddings on device. Taken together, these results indicate that multimodality is a practical way to move from post-hoc engagement analysis to actionable, real-time interventions in classroom ITS deployments, and they open the door to future work on learning intervention policies, fairness across student subgroups, and extension to multi-student settings.

References

[1] VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist, 46*(4), 197–221.

[2] Koedinger, K. R., Corbett, A. T., & Perfetti, C. (2012). The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science, 36*(5), 757–798.

[3] D'Mello, S. K., & Kory, J. (2015). A review and meta-analysis of multimodal affect detection systems. *ACM Computing Surveys, 47*(3), 1–36.

[4] Whitehill, J., Serpell, Z., Lin, Y.-C., Foster, A., & Movellan, J. R. (2014). The faces of engagement: Automatic recognition of student engagement from facial expressions. *IEEE Transactions on Affective Computing, 5*(1), 86–98.

[5] Baker, R. S. J. d., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining, 1*(1), 3–17.

[6] Pardos, Z. A., & Heffernan, N. T. (2010). Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization* (pp. 255–266).

[7] Ochoa, X., & Worsley, M. (2016). Augmenting learning analytics with multimodal sensory data. *Journal of Learning Analytics, 3*(2), 213–219.

[8] Calvo, R. A., & D'Mello, S. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing, 1*(1), 18–37.

[9] Pantic, M., & Rothkrantz, L. J. M. (2000). Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(12), 1424–1445.

[10] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2625–2634).

[11] Bosch, N., D'mello, S. K., Ocumpaugh, J., Baker, R. S., & Shute, V. (2016). Using video to automatically detect learner affect in computer-enabled classrooms. *ACM Transactions on Interactive Intelligent Systems (TiiS), 6*(2), 1-26.

[12] Grafsgaard, J. F., Wiggins, J. B., Boyer, K. E., Wiebe, E. N., & Lester, J. C. (2014). Predicting learning and affect from multimodal data streams in task-oriented tutorial dialogue. *International Journal of Artificial Intelligence in Education, 24*(1), 33–70.

[13] Siemens, G., & Baker, R. S. (2012). Learning analytics and educational data mining: Towards communication and collaboration. In *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (pp. 252–254).

[14] Beck, J. E., & Gong, Y. (2013). Wheel-spinning: Students who fail to master a skill. In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Proceedings of the 16th International Conference on Artificial Intelligence in Education* (pp. 431–440).

[15] Papamitsiou, Z., & Economides, A. A. (2014). Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Educational Technology & Society, 17*(4), 49–64.

[16] D'Mello, S., Dieterle, E., & Duckworth, A. (2017). Advanced, analytic, automated (AAA) measurement of engagement during learning. *Educational psychologist, 52*(2), 104-123.

[17] Blikstein, P., & Worsley, M. (2016). Multimodal learning analytics and education data mining: Using computational technologies to measure complex learning tasks. *Journal of Learning Analytics, 3*(2), 220–238.

[18] D'Mello, S., & Graesser, A. (2010). Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Modeling and User-Adapted Interaction, 20*(2), 147–187.

[19] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning* (pp. 689–696).

[20] Kizilcec, R. F., Piech, C., & Schneider, E. (2013). Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. In *Proceedings of the 3rd International Conference on Learning Analytics and Knowledge* (pp. 170–179).

[21] Baker, R. S., Goldstein, A. B., & Heffernan, N. T. (2011). Detecting learning moment-by-moment. *International Journal of Artificial Intelligence in Education, 21*(1-2), 5-25.

How to cite this article: Margrit Betke and Amelia Harper (2023). Multimodal Early Prediction of Student Problem Outcomes from Vision and Interaction Logs in an Intelligent Math Tutor. *Bulletin of Computer and Data Sciences*, 4(2), 30-42. DOI: 10.71448/bcds2342-3

Received: 09/06/2023 **Revised:** 28/06/2023 **Accepted:** 29/07/2023 **Publish:** 30/08/2023