Towards Explainable Student Group Collaboration Assessment Models Using Temporal Representations of Individual Student Roles

Anirudh Som
Center for Vision Technologies
SRI International
anirudh.som@sri.com

Sujeong Kim
Center for Vision Technologies
SRI International
sujeong.kim@sri.com

Bladimir Lopez-Prado
Center for Education
Research and Innovation
SRI International
bladimir.lopez-prado@sri.com

Svati Dhamija
Center for Vision Technologies
SRI International
svati.dhamija@sri.com

Nonye Alozie
Center for Education
Research and Innovation
SRI International
maggie.alozie@sri.com

Amir Tamrakar
Center for Vision Technologies
SRI International
amir.tamrakar@sri.com

ABSTRACT

Collaboration is identified as a required and necessary skill for students to be successful in the fields of Science, Technology, Engineering and Mathematics (STEM). However, due to growing student population and limited teaching staff it is difficult for teachers to provide constructive feedback and instill collaborative skills using instructional methods. Development of simple and easily explainable machine-learning-based automated systems can help address this problem. Improving upon our previous work, in this paper we propose using simple temporal-CNN deep-learning models to assess student group collaboration that take in temporal representations of individual student roles as input. We check the applicability of dynamically changing feature representations for student group collaboration assessment and how they impact the overall performance. We also use Grad-CAM visualizations to better understand and interpret the important temporal indices that led to the deep-learning model's decision.

Keywords

K-12, Education, Collaboration Assessment, Explainable, Deep-Learning, CNN, Grad-CAM, Cross-modal Analysis.

1. INTRODUCTION

Collaboration is considered a crucial skill, that needs to be inculcated in students early on for them to excel in STEM fields [24, 6]. Traditional instruction-based methods [14, 7] can often make it difficult for teachers to observe several student groups and identify specific behavioral cues that contribute or detract from the collaboration effort [20, 15, 25]. This has resulted in a surge in interest to develop machine-learning-based automated systems to assess student group collaboration [17, 11, 12, 8, 1, 9, 26, 23, 21, 4, 27, 22].

In our earlier work we developed a multi-level, multi-modal conceptual model that serves as an assessment tool for individual student behavior and group-level collaboration quality [2, 3]. Using the conceptual model as a reference, in a different paper we developed simple MLP deep-learning models that predict student group collaboration quality from histogram representations of individual student roles [22]. Please refer to the following papers for more information and for the illustration of the conceptual model [2, 3, 22]. Despite their simplicity and effectiveness, the MLP models and histogram representations lack explainability and insight into the important student dynamics. To address this, in this paper we focus on using simple temporal-CNN deep learning models to check the scope of dynamically changing temporal representations for student group collaboration assessment. We also use Grad-CAM visualizations to help identify important temporal instances of the task performed and how they contribute towards the model's decision.

Paper Outline: Section 2 provides necessary background on the different loss functions used, dataset description and the temporal features extracted. Section 3 describes the experiments and results. Section 4 concludes the paper.

2. BACKGROUND

2.1 Cross-Entropy Loss Functions

The categorical-cross-entropy loss is the most commonly used loss function to train deep-learning models. For a classification problem with C classes, let us denote the input variables as $\mathbf{x}$, ground-truth label vector as $\mathbf{y}$ and the predicted probability distribution as $\mathbf{p}$. Given a training sample $(\mathbf{x}, \mathbf{y})$, the categorical-cross-entropy (CE) loss is defined as

Table 1: Coding rubric for Level A and Level B2.

Level A	Level B2
Effective [E]	Group guide/Coordinator [GG]
Satisfactory [S]	Contributor (Active) [C]
Progressing [P]	Follower [F]
Needs Improvement [NI]	Conflict Resolver [CR]
Working Independently [WI]	Conflict Instigator/Disagreeable [CI]
	Off-task/Disinterested [OT]
	Lone Solver [LS]

Table 2: Inter-rater reliability (IRR) measurements.

Level	Average Agreement	Cohen's Kappa
A	0.7046	0.4908
B2	0.6741	0.5459

$$\text{CE}_{\mathbf{x}}(\mathbf{p}, \mathbf{y}) = -\sum_{i=1}^{C} \mathbf{y}_i \log(\mathbf{p}_i) \tag{1}$$

Here, $\mathbf{p}_i$ denotes the predicted probability of the i-th class. Note, both $\mathbf{y}$ and $\mathbf{p}$ are of length C, with $\sum_i \mathbf{y}_i = \sum_i \mathbf{p}_i = 1$. From Equation 1, it's clear that for imbalanced datasets the learnt weights of the model will be biased towards classes with the most number of samples in the training set. Additionally, if the label space exhibits an ordered structure, the categorical-cross-entropy loss will only focus on the predicted probability of the ground-truth class while ignoring how far off the incorrectly predicted sample actually is. These limitations can be addressed to some extent by using the ordinal-cross-entropy (OCE) loss function [22], defined in Equation 2.

$$\begin{aligned} \text{OCE}_{\mathbf{x}}(\mathbf{p}, \mathbf{y}) &= -(1+w)\sum_{i=1}^{C} \mathbf{y}_i \log(\mathbf{p}_i) \\ w &= |\text{argmax}(\mathbf{y}) - \text{argmax}(\mathbf{p})| \end{aligned} \tag{2}$$

Here, $(1+w)$ represents the weighting variable, argmax returns the index of the maximum valued element and $|.|$ returns the absolute value. When training the model, $w = 0$ for correctly classified training samples, with the ordinal-cross-entropy loss behaving exactly like the categorical-cross-entropy loss. However, for misclassified samples the ordinal-cross-entropy loss will return a higher loss value. The increase in loss is proportional to how far away a sample is misclassified from its ground-truth class label.

2.2 Dataset Description

We collected audio and video recordings from 15 student groups, across five middle schools. Out of the 15 groups, 13 groups had 4 students, 1 group had 3 students, and 1 group had 5 students. The student volunteers completed a brief survey that collected their demographic information and other details, e.g., languages spoken, ethnicity and comfort levels with science concepts. Each group was tasked with completing 12 open-ended life science and physical science tasks, which required them to construct models of different science phenomena as a team. They were given one hour to complete as many tasks possible, which resulted in 15 hours of audio and video recordings. They were provided logistic and organization instructions but received no help in group dynamics, group organization, or task completion.

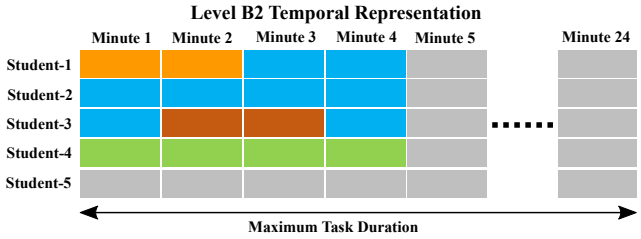

Figure 1: Level B2 temporal representation for a group having only 4 students and finishing the assigned task in 4 minutes. Colored cells illustrate the different Level B2 codes as described in Table 1, and the gray cells represent empty or unassigned codes.

Next, the data recordings were manually annotated by education researchers at SRI International. For the rest of the paper we will refer to them as coders/annotators. In our hierarchical conceptual model [2, 3], we refer to the collaboration quality annotations as Level A and individual student role annotations as Level B2. The coding rubric for these two levels is described in Table 1. Both levels were coded by three annotators. They had access to both audio and video recordings and used ELAN (an open-source annotation software) to annotate. A total of 117 tasks were coded by each annotator, with the duration of each task ranging from 5 to 24 minutes. Moderate-agreement was observed across the coders as seen from the inter-rater reliability measurements in Table 2.

Level A codes represent the target label categories for our classification problem. To determine the ground-truth Level A code, the majority vote (code) across the three annotators was used as the ground-truth. For cases where a majority was not possible, we used the Level A code ordering depicted in Table 1 to determine the median as ground-truth of the three codes. For example, if the three coders assigned *Satisfactory, Progressing, Needs Improvement* for the same task then *Progressing* would be used as the ground-truth label. Note, we did not observe a majority Level A code for only 2 tasks. To train the machine learning models we only had 351 data samples (117 tasks $\times$ 3 coders).

2.3 Temporal Representation

In our dataset, the longest task was little less than 24 minutes, due to which the length for all tasks was also set to 24 minutes. Level B2 was coded using fixed-length 1 minute segments, as illustrated in Figure 1. Due to its fixed-length nature, we assigned an integer value to each B2 code, i.e., the seven B2 codes were assigned values from 1 to 7. The value 0 was used to represent segments that were not assigned a code. For example, in Figure 1 we see a group of 4 students completing a task in just 4 minutes, represented by the colored cells. The remaining 20 minutes and the 5^{th} student is assigned a value zero, represented by the gray cells. Thus for each task, Level B2 temporal features will have a shape 24×5, with 24 representing number of minutes and 5

representing number of students in the group.

Baseline Histogram Representation: We compare the performance of the temporal representations against simple histogram representations [22]. The histogram representations were created by pooling over all the codes observed over the duration of the task and across all the students. Note, only one histogram was generated per task, per group. Once the histogram is generated we normalize it by dividing by the total number of codes in the histogram. Normalizing the histogram removes the temporal aspect of the task. For example, if group-1 took 10 minutes to solve a task and group-2 took 30 minutes to solve the same task, but both groups were assigned the same Level A code despite group-1 finishing the task sooner. The raw histogram representations of both these groups would look different due to the difference in number of segments coded. However, normalized histograms would make them more comparable. Despite the normalized histogram representation being simple and effective, it fails to offer any insight or explainability.

3. EXPERIMENTS

Network Architecture: For the temporal-CNN deep learning model we used the temporal ResNet architecture described in [28]. The ResNet architecture uses skip connections between each residual block to help avoid the vanishing gradient problem. It has shown state-of-the-art performance in several computer vision applications [10]. Following [28], our ResNet model consists of three residual blocks stacked over one another, followed by a global-average-pooling layer and a softmax layer. The number of filters for each residual block was set to 64, 128, 128 respectively. The number of learnable parameters for the B2 temporal representations is 506949. We compare the performance of the ResNet model to the MLP models described in our previous work. Interested readers should refer to [22] for more information about the baseline MLP model that was used with the histogram representation.

Training and Evaluation Protocol: All models were developed using Keras with TensorFlow backend [5]. We used the Adam optimizer [13] and trained all models for 500 epochs. The batch-size was set to one-tenth of the number of training samples during any given training-test split. We optimized over the Patience and Minimum-Learning-Rate hyperparameters, that were set during the training process. We focused on these as they significantly influenced the model's classification performance. The learning-rate was reduced by a factor of 0.5 if the loss did not change after a certain number of epochs, indicated by the Patience hyperparameter. We saved the best model that gave us the lowest test-loss for each training-test split. We used a round-robin leave-one-group-out cross validation protocol. This means that for our dataset consisting of g student groups, for each training-test split we used data from $g-1$ groups for training and the left-out group was used as the test set. This was repeated for all g groups and the average result was reported. For our experiments $g = 14$ though we have temporal representations from 15 student groups. This is because all samples corresponding to the *Effective* class were found only in one group. Due to this reason and because of our cross-validation protocol we do not see any test samples for the *Effective* class.

Table 3: Weighted precision, weighted recall and weighted F1-score Mean±Std for the best MLP and ResNet models under different settings.

Feature	Classifier	Weighted Precision	Weighted Recall	Weighted F1-Score
B2 Histogram	SVM	84.45±13.43	73.19±16.65	76.92±15.39
	MLP - Cross-Entropy Loss	83.72±16.50	86.42±10.44	84.40±13.85
	MLP - Cross-Entropy Loss + Class-Balancing	83.93±17.89	85.29±14.37	84.16±16.23
	MLP - Ordinal-Cross-Entropy Loss	86.96±14.56	88.78±10.36	**87.03±13.16**
	MLP - Ordinal-Cross-Entropy Loss + Class-Balancing	86.73±14.43	88.20±9.66	86.60±12.54
B2 Temporal	ResNet - Cross-Entropy Loss	84.75±13.21	83.10±11.92	82.72±12.74
	ResNet - Cross-Entropy Loss + Class-Balancing	84.03±15.13	83.28±11.42	82.97±12.84
	ResNet - Ordinal-Cross-Entropy Loss	85.24±15.68	87.23±10.52	85.56±13.38
	ResNet - Ordinal-Cross-Entropy Loss + Class-Balancing	84.34±15.75	87.88±11.22	**85.68±13.58**

3.1 Temporal vs Histogram Representations

Here, we compare the performance of the ResNet and MLP models. Using the weighted F1-score performance, Table 3 summarizes the best performing ResNet and MLP models for the different feature-classifier variations. The table also provides the weighted precision and recall metrics. Bold values in the table represent the best classifier across the different feature-classifier settings. The ordinal-cross-entropy loss with or without class-balancing shows the highest weighted F1-score performance for both feature types. Here, class-balancing refers to weighting each data sample by a weight that is inversely proportional to the number of data samples corresponding to that sample's ground-truth label.

At first glance, the ResNet models perform slightly less than the MLP models. This could easily lead us to believe that simple histogram representations are enough to achieve a higher classification performance than the corresponding temporal representations. However, despite the performance differences, the temporal features and ResNet models help better explain and pin-point regions in the input feature space that contribute the most towards the model's decision. This is important if one wants to understand which student roles are most influential in the model's prediction. We will go over this aspect in more detail in the next section.

3.2 Grad-CAM Visualization

Grad-CAM uses class-specific gradient information, flowing into the final convolutional layer to produce a coarse localization map that highlights the important regions in the input feature space [19]. It is primarily used as a post hoc analysis tool and is not used in any way to train the model. Figure 2 illustrates how Grad-CAM can be used for our classification problem. We show two different samples from the Satisfactory, Progressing and Needs Improvement classes respectively. Each sample shows a group consisting of 4 students that completed the task in 5 to 8 minutes. Technically one can obtain C Grad-CAM maps for a C-class classification problem. Here, the samples shown correspond to the class predicted by the ResNet model, which is also the ground-truth class. It's clear how the Grad-CAM highlights regions in the input feature space that contributed towards the correct prediction. For instance, in the Needs Improvement examples, the Grad-CAM map shows the highest weight on the fourth minute. At that time for the first example, the codes for three of the students become Off-task/Disinterested. Similarly, for the second example we notice three of the students become Lone Solvers and the

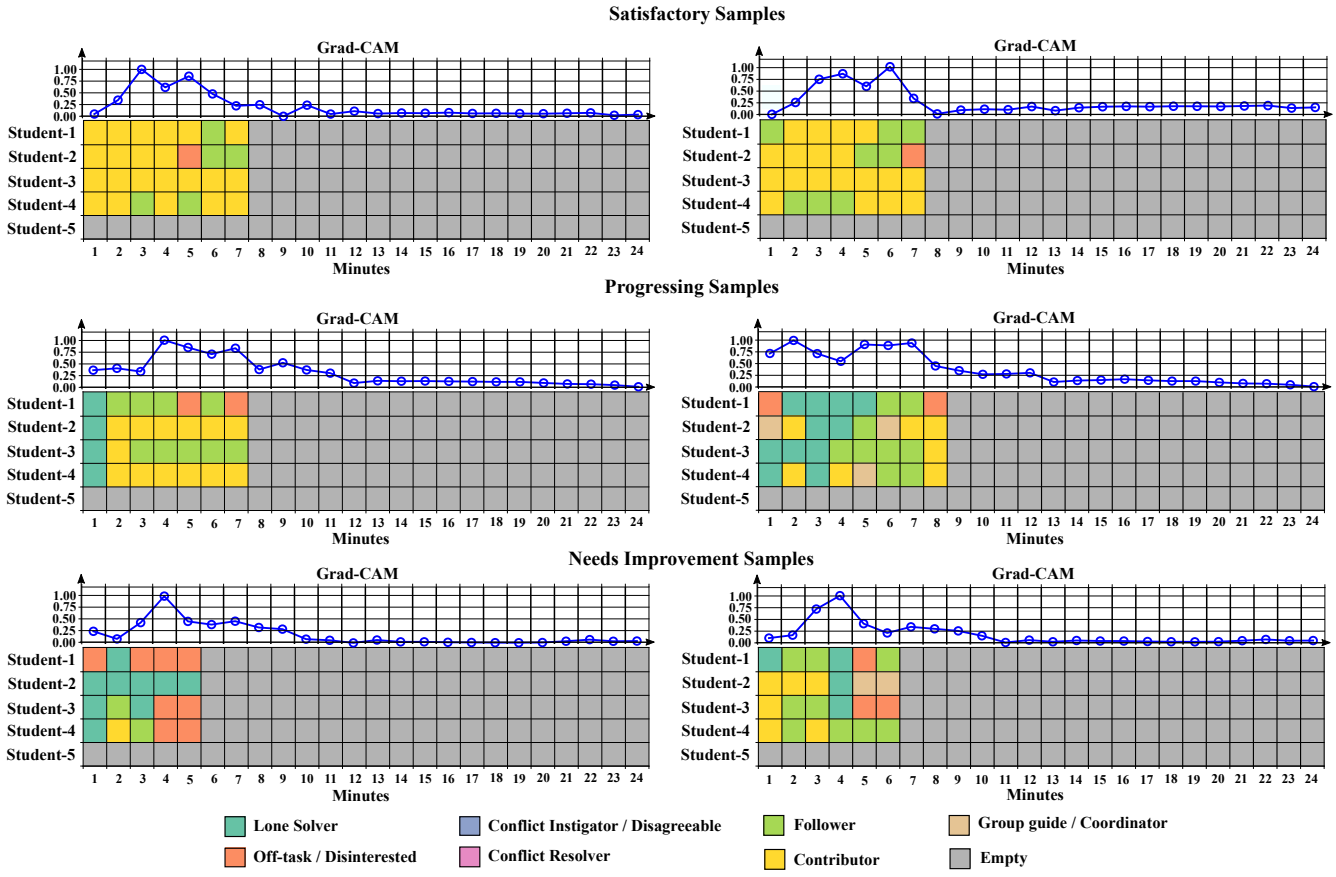

Figure 2: Grad-CAM visualization for two different temporal samples from different Level-A classes.

fourth student becomes a Follower. This is in stark contrast to the minute before when two of the students were Followers and the other two were Contributors. We also notice less importance being given to the Empty codes. These changes in roles and the Grad-CAM weights across the task make sense and help promote explainability in our deep learning models.

4. CONCLUSION

In this paper we proposed using simple temporal representations of individual student roles together with temporal ResNet deep-learning architectures for student group collaboration assessment. Our objective was to develop more explainable systems that allow one to understand which instances in the input feature space led to the deep-learning model's decision. We suggested use of Grad-CAM visualization along the temporal dimension to assist in locating important time instances in the task performed. We compared the performance of the proposed temporal representations against simpler histogram representations from our previous work [22]. While histogram representations can help achieve high classification performance, they do not offer the same key insights that one can get using the temporal representations.

Limitations and Future Work: The visualization tools and findings discussed in this paper can help guide and shape future work in this area. Having said that our approach can be further extended and improved in several ways. For example, we only discuss Grad-CAM maps along the temporal dimensions. This only allows us to identify important temporal instances of the task but does not focus on the important student interactions. The current setup does not tell us which subset of students are interacting and how that could affect the overall group dynamic and collaboration quality. To address this we intend on exploring other custom deep-learning architectures and feature representation spaces. We also plan on using other tools like LIME [18] and SHAP [16]. These packages compute the importance of the different input features and help towards better model explainability and interpretability. Also we only focused on mapping deep learning models from individual student roles to overall group collaboration. In the future we intend on exploring other branches in the conceptual model, described in [2, 3]. We also plan on developing recommendation systems that can assist and guide students to improve themselves by suggesting what they need to take on. The same system could also be tweaked specifically for teachers to give them insight on how different student interactions could be improved to facilitate better group collaboration.

5. ACKNOWLEDGEMENT

This work was supported in part by NSF grant number 2016849.

6. REFERENCES

[1] G. Alexandron, J. A. Ruipérez-Valiente, and D. E. Pritchard. Towards a general purpose anomaly detection method to identify cheaters in massive open online courses. 2020.

[2] N. Alozie, S. Dhamija, E. McBride, and A. Tamrakar. Automated collaboration assessment using behavioral analytics. 2020.

[3] N. Alozie, E. McBride, and S. Dhamija. Collaboration conceptual model to inform the development of machine learning models using behavioral analytics. 2020.

[4] A. R. Anaya and J. G. Boticario. Application of machine learning techniques to analyse student interactions and improve the collaboration process. *Expert Systems with Applications*, 38(2):1171–1181, 2011.

[5] F. Chollet et al. Keras. https://keras.io, 2015.

[6] W. R. Daggett and D. S. GendroO. Common core state standards initiative. *International center*, 2010.

[7] N. Davidson and C. H. Major. Boundary crossings: Cooperative learning, collaborative learning, and problem-based learning. *Journal on excellence in college teaching*, 25, 2014.

[8] C. Genolini and B. Falissard. Kml: A package to cluster longitudinal data. *Computer methods and programs in biomedicine*, 104(3):e112–e121, 2011.

[9] Z. Guo and R. Barmaki. Collaboration analysis using object detection. In *EDM*, 2019.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] K. Huang, T. Bryant, and B. Schneider. Identifying collaborative learning states using unsupervised machine learning on eye-tracking, physiological and motion sensor data. *International Educational Data Mining Society*, 2019.

[12] J. Kang, D. An, L. Yan, and M. Liu. Collaborative problem-solving process in a science serious game: Exploring group action similarity trajectory. *International Educational Data Mining Society*, 2019.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] J. S. Krajcik and P. C. Blumenfeld. *Project-based learning*. na, 2006.

[15] M. L. Loughry, M. W. Ohland, and D. DeWayne Moore. Development of a theory-based assessment of team member effectiveness. *Educational and psychological measurement*, 67(3):505–524, 2007.

[16] S. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[17] J. M. Reilly and B. Schneider. Predicting the quality of collaborative problem solving through linguistic analysis of discourse. *International Educational Data Mining Society*, 2019.

[18] M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[20] K. A. Smith-Jentsch, J. A. Cannon-Bowers, S. I. Tannenbaum, and E. Salas. Guided team self-correction: Impacts on team mental models, processes, and effectiveness. *Small Group Research*, 39(3):303–327, 2008.

[21] A. Soller, J. Wiebe, and A. Lesgold. A machine learning approach to assessing knowledge sharing during collaborative learning activities. 2002.

[22] A. Som, S. Kim, B. Lopez-Prado, S. Dhamija, N. Alozie, and A. Tamrakar. A machine learning approach to assess student group collaboration using individual level behavioral cues. In *European Conference on Computer Vision Workshops*, pages 79–94. Springer, 2020.

[23] D. Spikol, E. Ruffaldi, and M. Cukurova. Using multimodal learning analytics to identify aspects of collaboration in project-based learning. Philadelphia, PA: International Society of the Learning Sciences., 2017.

[24] N. L. States. *Next generation science standards: For states, by states*. The National Academies Press, 2013.

[25] S. Taggar and T. C. Brown. Problem-solving team behaviors: Development and validation of bos and a hierarchical factor structure. *Small Group Research*, 32(6):698–726, 2001.

[26] L. Talavera and E. Gaudioso. Mining student data to characterize similar behavior groups in unstructured collaboration spaces. In *Workshop on artificial intelligence in CSCL. 16th European conference on artificial intelligence*, pages 17–23. Citeseer, 2004.

[27] H. Vrzakova, M. J. Amon, A. Stewart, N. D. Duran, and S. K. D'Mello. Focused or stuck together: Multimodal patterns reveal triads' performance in collaborative problem solving. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 295–304, 2020.

[28] Z. Wang, W. Yan, and T. Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.