The Company You Keep: Refining Neural Epistemic Network Analysis

Zheng Fang*
Monash University
Clayton, VIC, Australia
zheng.fang@monash.edu

Weiqing Wang
Monash University
Clayton, Australia
Teresa.Wang@monash.edu

Guanliang Chen
Monash University
Clayton, Australia
Guanliang.Chen@monash.edu

Zachari Swiecki
Monash University
Clayton, Australia
Zach.Swiecki@monash.edu

Abstract

Collaborative problem-solving (CPS) is defined as an inherently sociocognitive phenomena. Despite this, extant learning analytic techniques tend to focus on either the social or cognitive aspects without explicitly considering their interaction. Prior work developed Neural Epistemic Network Analysis (NENA), which used a combination of deep learning methods to simultaneously model the social and cognitive aspects of CPS; however, the method had several limitations. The refined version of NENA presented here addresses these limitations by (a) introducing a simplified autoencoder deep learning architecture; (b) using a combination of social and epistemic networks as input to preserve interpretability in terms of social and cognitive factors; and (c) introducing an isometry loss function to ensure downstream statistical tests are meaningful. We found that the refined version of NENA is able to achieve high performance on criteria we would expect from a network analytic technique in the context of learning analytics: interpretability, goodness of fit, orthogonality and isometry; and discriminatory power. We also demonstrated that this method was comparable in performance to a more traditional learning analytic technique, Epistemic Network Analysis (ENA), while providing information that ENA did not. The results suggest that NENA could be a useful method for exploring the cognitive interactions of a given individual's social network and thus the influences their network exerts upon them.

CCS Concepts

• Computing methodologies → Artificial intelligence; **• Applied computing → Collaborative learning**; **• Human-centered computing → Empirical studies in visualization**; **Social network analysis**.

*Corresponding Author

LAK 2025, Dublin, Ireland

ACM ISBN 979-8-4007-0701-8/25/03
https://doi.org/10.1145/3706468.3706497

Keywords

Collaborative Problem-Solving, Epistemic Network Analysis, Social Network Analysis, Graph Neural Networks

ACM Reference Format:
Zheng Fang, Weiqing Wang, Guanliang Chen, and Zachari Swiecki. 2025. The Company You Keep: Refining Neural Epistemic Network Analysis. In *LAK25: The 15th International Learning Analytics and Knowledge Conference (LAK 2025), March 03–07, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3706468.3706497

1 Introduction

Collaborative problem solving (CPS) [8] is a critical skill in many domains, requiring individuals to work together to achieve a common goal. This process involves both cognitive and social dimensions [2], where participants share information, negotiate understanding, and build on each other's ideas. Effective CPS not only depends on individual problem-solving skills but also on the ability to interact and communicate effectively within a group [4, 5, 16].

Methods for analyzing the social and cognitive aspects of CPS are well-developed, but methods for simultaneously modeling both are not. Epistemic Network Analysis (ENA) [19, 20] is a well-established technique for analyzing the cognitive aspects of collaboration by modeling relationships between coded elements (e.g., dialogue acts, behaviors) and visualizing how participants' understanding and strategies evolve over time. ENA uses co-occurrence of codes along with dimensionality reduction techniques like Singular Value Decomposition (SVD) to transform cognitive interactions into diagrams for visualization [3, 18]. While ENA has been widely used to analyze CPS in various settings, such as educational environments and military training, it does not explicitly model the social interactions that are equally vital in collaborative settings [22, 23].

Previous work introduced Neural Epistemic Network Analysis (NENA) to address this limitation by integrating ENA with Graph Neural Networks (GNNs), which can naturally combine cognitive and social information [26]. However, the initial version of NENA had limitations in terms of (a) its methodological architecture, (b) its interpretability, and (c) its appropriateness for conducting statistical tests [6]. Here, we introduce an improved version of NENA that overcomes these limitations. In this new version of NENA, co-occurrence data from ENA is used as node features, interactions between the participants are used as the weighted edges within a GNN framework, and an autoencoder [14] is used to capture complex relationships in the data.

Our results on a widely-used CPS dataset show that NENA is comparable to traditional ENA on several key metrics while also providing distinct information. By integrating GNNs with ENA, NENA has the potential to advance the theoretical and practical understanding of CPS. This interdisciplinary approach provides a more holistic view of CPS, highlighting the importance of both social and cognitive interactions in achieving effective collaboration. Through this research, we aim to contribute to the fields of learning analytics and educational research, offering new tools and methodologies for analyzing and understanding collaborative processes.

2 Background

Collaboration analytics (CA) has emerged as a pivotal area of research in education, focusing on understanding and enhancing the dynamics of collaboration [12]. A key component of CA is collecting low-level data from collaborative interactions, organizing that data into constructs relevant to collaboration, and then applying analytic techniques. This process involves capturing detailed data on how participants interact, communicate, and work together. By structuring this data into meaningful constructs, researchers can apply sophisticated analytical methods to uncover patterns, dynamics, and insights that inform the understanding and enhancement of collaborative processes.

A widely used technique for CA is Epistemic Network Analysis (ENA) [17, 19], which quantifies and visualizes the relationships among coded elements within qualitative data—such as texts or actions—transforming these into network graphs. The method begins by categorizing data into codes representing critical concepts or interactions and then counts the frequency of code co-occurrences. Through dimensional reduction, ENA represents these relationships in a low-dimensional space, yielding network graphs and associated embeddings, or *ENA scores*, that show the connections between concepts or ideas people made during collaboration.

Similarly, Social Network Analysis (SNA) [9] is a technique that specifically focuses on the social aspect CPS, examining the patterns of interaction among individuals within a group. SNA models the network of social interactions, where nodes represent individuals and edges indicate the frequency and intensity of interactions. This approach provides insights into the social structure of collaboration, such as network density, centrality of participants, and structural equivalence, among other characteristics.

This clear delineation between the cognitive and social analytic technique for CA highlights a critical gap in the analysis of CPS. Cognitive methods like ENA are adept at mapping out knowledge and idea exchanges but may not fully capture how these cognitive processes are influenced by social interactions. In contrast, methods like SNA excel at analyzing interaction patterns but tend not to probe into the substantive nature of these interactions or how they contribute to collective cognition and problem-solving. This situation underscores the need for an integrated approach that merges the strengths of both cognitive and social analyses to provide a more holistic understanding of CPS.

Few attempts have been made to meaningfully integrate both the social and cognitive dimensions of CPS within the same analysis. Social epistemic Network Signature (SENS) [7] is analytical framework that specifically focuses on the both cognitive and social aspect of CPS. It combines metrics from SNA and ENA to explore how social networks influence and are influenced by collaborative activities. Swiecki and Shaffer[24] extended SENS via that Integrated Social-Epistemic Network Signature (iSENS). In iSENS, weighted network graphs are constructed where nodes represent individual participants, and edges denote the frequency of their interactions, mirroring the social dynamics of communication. Uniquely, the positioning of these nodes is not arbitrary but is instead determined by their ENA scores, situating individuals within the network based on the cognitive aspects of their contributions. This positioning ensures that iSENS networks concurrently reflect the social structure through the relational edges between participants and the cognitive structure through the spatial arrangement of nodes.

Despite its advancements, iSENS is limited due to its reliance on somewhat simplistic mathematics for integrating social and cognitive information. Specifically, the model's approach to linearly updating the positions of nodes based on interaction frequency can inadvertently overemphasize the contributions of individuals who are more active but may not necessarily provide novel insights into the collaboration. Recently, [Blinded] proposed and tested a new method for modelling the sociocognitive nature of CPS, Neural Epistemic Network Analysis (NENA) [6]. NENA combines ENA with a powerful deep learning technique, Graph Neural Networks (GNNs) [26], a class of neural networks designed to process data structured as graphs, effectively capturing the relationships and interactions between nodes.

The original version of NENA defines a graph structure with individuals as nodes, levels of communication as edges, and code co-occurrence vectors as node features. It uses Graph Convolutional Networks (GCNs) [1, 15] to generate high-dimensional embeddings for each node in a latent space that distinguishes nodes in terms of both social and cognitive information. Next, NENA uses a *distillation* process [6], where a separate neural network is used to learn a rotation matrix that, when multiplied by the original graph data, can approximate the high-dimensional embeddings. Finally, the co-occurrence vectors from ENA are multiplied by this rotation matrix to produce low-dimensional embeddings, and co-occurrence networks are plotted in this space using the same optimization routine used by the ENA algorithm. The upshot of the method is that individuals are distinguished in the low dimensional space in terms of both who they communicated with and what they communicated about.

The previous study compared the results of NENA to traditional ENA in terms of: (1) the qualitative and quantitative similarity of the produced embedding spaces; (2) the goodness of fit between the networks and embeddings; (3) the orthogonality of the dimensions of the embedding spaces; (4) the preservation of original distances between the points in the embedding spaces (isometry); and the ability to distinguish between known subpopulations in the data. The results showed that NENA produced dimensions with different interpretations, high goodness of fit, and better predictive power. However, the results were less orthogonal and isometric compared to ENA.

Despite promising preliminary results, the initial version of NENA has several important drawbacks that we seek to address in this work. First, the use of the distillation method introduced an additional deep learning approach to the method , and the model was sensitive to various hyperparamaters, making it less generalisable and harder to implement.

Second, at a high-level, the previous version of NENA works by using GNNs and distillation to learn a *rotation matrix*, that when multiplied by the original co-occurrence data from ENA produces new *NENA scores* in a low dimensional space. While the GNN structure ensured that the rotation matrix captured information about social and cognitive connections, applying this matrix to only to the ENA co-occurrence (cognitive) data means that the method occludes the impact of the social information in the final output. In other words, NENA scores are positioned in the space in part because of the social information in the data, but it is not possible to determine what that social information is.

Finally, ENA outperformed the original implementation of NENA in terms of orthogonality and isometry. While orthogonal dimensions are not a necessary feature of rotation matrices, they can aid in interpretability. And while many dimensional reduction techniques fail to preserve isometry, they are primarily used for visualisation and clustering [11, 13] — tasks were preserving distances is less important. In contrast, ENA is often used to compare networks via their embeddings in the low dimensional space using statistical tests that measure distance in the space. This purpose requires that distances between embeddings are meaningful, not arbitrary. As the original NENA approach failed to preserve a high level of isometry, any advantages the method shows in distinguishing between networks are suspect.

The improved version of NENA addresses the first limitation by removing the distillation process, instead using an autoencoder to generate latent embeddings for the graph nodes—which in this case represent participants in CPS activities—in an end-to-end manner. GNN autoencoders are a type of neural network architecture designed to encode graph-structured data into a latent space and then decode it back to the original structure [26, 27]. This approach simplifies the overall architecture and reduces the need for complex hyperparameters, improving efficiency and generalisability. To address the second limitation, the method generates a rotation matrix that can be multiplied by a representation of the original data to yield low dimensional embeddings (NENA scores). Unlike the previous version of NENA, which used the ENA co-occurrences as the original data input—the improved version uses a combination of ENA and SNA networks as the original data input, which ensures that both the input and rotation matrix contain cognitive and social information. In turn, the resulting dimensions of the embedding space can be interpreted explicitly in terms of cognitive and social information. To address the third limitation, we incorporate a new loss function into the architecture that seeks to preserve the distances between the original data points after rotation.

We evaluated this improved version of NENA by comparing it ENA in terms of (a) the interpretation of the resulting embedding spaces; (b) the goodness of fit between the network representations and embeddings; (c) orthogonality of the dimensions of the embedding spaces; isometry of the data points before and after rotation; (d) ability to statistically distinguish between subpopulations in the data; and (e) the ability to distinguish individuals in the data in terms of their sociocognitive behaviour.

3 Methods

3.1 Data

Following [6], we compare ENA and NENA using a CPS dataset collected as part of the Tactical Decision Making Under Stress (TADMUS) study involving 16 Anti-Defense Warfare (ADW) teams who participated in four simulated training scenarios [10]. This study aimed to assess the impact of team training and a decision support system on team performance. Each team comprised six members: two in command roles and four in support roles, with both sets of roles focused on maintaining tactical awareness in simulated littoral warfare scenarios. In these scenarios, teams needed to identify whether other vessels detected via radar, referred to as "tracks," were friend or foe and respond accordingly.

The study divided teams into two conditions. The control group used standard ADW support tools, while the experimental group utilized an enhanced decision support system designed to reduce cognitive load and promote tactical awareness. Additionally, the experimental teams received specialized computer-based and face-to-face training to improve critical thinking and teamwork skills.

The dataset includes transcripts of team conversations across the four different ADW scenarios, segmented by turn of talk, totaling 12,028 turns. A set of qualitative codes representing epistemic actions in the ADW context was developed, validated, and applied to the data, as part of previous research by Swiecki and colleagues [23]. These codes describe the ADW "detect-to-engage sequence," which involves teams (a) passing and seeking information about track detection, behavior, and identity, (b) assessing track threat levels, and (c) recommending and executing tactical actions against threatening tracks (see Table 1). We segmented the data for analysis according to speaker, team, and training scenario. As their were four scenarios, each speaker could appear up to four times in the final ENA/NENA datasets (see below). In total, the data included 422 unique units of analysis—i.e., unique combinations of speaker, team, and scenario.

3.2 Epistemic Network Analysis

As a benchmark for comparison, we applied ENA to the TADMUS dataset. The ENA algorithm processes the data by moving a fixed-size window over each turn of talk, counting the co-occurrences between codes within the window for each unit of analysis, and representing these co-occurrences as high-dimensional vectors. Dimensional reduction using SVD is then performed on these vectors to produce low-dimensional embeddings, known as ENA scores—each unit of analysis is associated with a corresponding network and ENA score. These units can be visually compared in the low-dimensional space using network graphs and statistically through their scores.

The SVD-based dimensional reduction in standard ENA creates an embedding space with orthogonal dimensions, maximizing the variance among the units of analysis. Researchers typically focus on and visualize the first two dimensions of this space, as they account for the most significant portion of variance while keeping the visualization straightforward. For this comparison, we used

Code	**Definition**	**Examples**
Detect/Identify	Talk about radar detection of a track or the identification of a track, (e.g., vessel type).	1) IR/EW NEW BEARING, BEARING 078 APQ120 CORRELATES TRACK 7036 POSSIBLE F-4 2) TIC/IDC TRACK 7023 NO MODES NO CODES
Track Behavior	Talk about kinematic data about a track or a track's location	1) AIR/IDS TRACK NUMBER 7021 DROP IN ALTITUDE TO 18 THOUSAND FEET 2) TAC/AIR TRACK 7031 THE HELO THAT WAS TAKING OFF THE OIL PLATFORM IS TURNED EAST
Assessment/Prioritization	Talk about whether a track is friendly or hostile, the threat level of a track, or indicating tracks of interest	1) TRACKS OF INTEREST 7013 LEVEL 5 7037 LEVEL 5 7007 LEVEL 4 TRACK 7020 LEVEL 5 AND 7036 LEVEL 5 2) CO, AYE. LET'S UPGRADE OUR THREAT LEVEL TO 6
Status Updates	Talk about procedural information, e.g., track responses to tactical actions, or talk about tactical actions taken by the team	1) TAO ID, STILL NO RESPONSE FROM TRACK 37, POSSIBLE PUMA HELO. 2) GOT HIM COVERED.
Seeking Information	Asking questions regarding track behavior, identification, or status.	1) TAO CO, WE'VE UPGRADED THEM TO LEVEL 7 RIGHT? 2) WHERE IS 23?
Recommendations	Recommending or requesting tactical actions	1) AIR/TIC RECOMMEND LEVEL THREE ON TRACK 7016 7022 2) GB, THIS IS GW, REQUEST PERMISSION TO TAKE TRACK 7022 WITH BIRDS HOSTILE AIR. THREAT LEVEL HIGH. RANGE 7NM.
Deterrent Orders	Giving orders meant to warn or deter tracks.	1) TIC AIR, CONDUCT LEVEL 2 WARNING ON 7037 2) AIR THIS IS TAC CONDUCT LEVEL ONE QUERY TRACK 7036
Defensive Orders	Giving orders to prepare ship defenses or engage hostile tracks	1) TAO/CO COVER 7016 WITH BIRDS 2) AIR KILL TRACK 7022 WITH BIRDS

Table 1: Qualitative codes, definitions, and examples.

a variation of the standard SVD procedure called *means rotation*, which sets the first dimension of the embedding space as the dimension that maximises between-group variance, where the two groups in this case are the control and experimental conditions in the TADMUS dataset. Subsequent dimensions are orthogonal to the means rotated dimension and determined via SVD (see [3] for more details.)

Mathematically, each dimension is represented as a linear combination of the original variables, which in this context are the co-occurrences between codes. The rotation involves multiplying the high-dimensional data by a set of weights to generate the low-dimensional embeddings. Importantly, this rotation process preserves the isometry of the points. The weights with large magnitudes are the most influential, and they help differentiate observations along the dimensions. ENA aids in interpreting these dimensions by aligning the nodes of each network and the midpoints of their edges as closely as possible with the weights used for the rotation. This alignment is achieved by minimizing the distance between the network centroids and the ENA scores via regression analysis [3].

The ENA model on the TADMUS data (a) produced one network and ENA score for each unique combination of speaker, team, and scenario in the data; (b) segmented the speaker data by team and scenario for co-occurrence identification; (c) used the the codes in Table 1 as nodes; and (d) a window size of five turns of talk.

3.3 Social Network Analysis

NENA combines both cognitive networks (ENA) and social networks (SNA). Here, we generated social networks for each unit of analysis by first segmenting the dataset by 'Scenario' and 'Team'. For each segment, the algorithm slides a window through the data rows—which correspond to turns of talk—treating each unit of analysis—i.e., each unique speaker—as a node in the social network. If two nodes co-occur within a window an edge is created between them with an initial weight of 1. If an edge already existed, the weight is incremented by 1, indicating multiple interactions between the two nodes. This process generates a directed graph with weighted edges that reflects the frequency of interactions between units of analysis that we represent as an adjacency matrix. Here, we used a window size of five to match the ENA window size.

3.4 Neural Epistemic Network Analysis

The new version of NENA uses a GNN autoencoder to generate embeddings for each unit of analysis in the data. In this setup, the nodes represent the unique combinations of speaker, team, and scenario, node features correspond to each unit's co-occurrence vector derived from ENA (cognitive component), and the weighted edges between nodes represent the frequency of communication between units (social component) (see 1). Broadly, NENA involves:

(1) Deriving co-occurrence vectors: These vectors represent the cognitive connections made by each unit of analysis and serve as node features within the GNN autoencoder. They are obtained by applying ENA to the data, as described above.
(2) Deriving the adjacency matrix: This matrix captures the social interactions of each unit of analysis and represents the network edges in the GNN autoencoder. The matrix is obtained by applying SNA to the data, as described above.
(3) Combining cognitive and social information: To reflect the interplay between social interactions and cognitive processes, we multiply the adjacency matrix (2) by the co-occurrence vectors (1) to produce a matrix where each row corresponds to a unit of analysis and the columns represent a given co-occurrence frequency weighted by each individual's frequency of social interaction.
(4) Encoding: The autoencoder consists of encoder and decoder components. The encoder uses layers of GCNConv which apply convolution operations to aggregate information from neighboring nodes and GATConv which use attention mechanisms to weigh the importance of neighboring nodes' features [25], to transform the input into a latent space. The goal of this transformation is to embed the units of analysis into a new space such that units who are sociocognitively similar are closer together in the space than those who are sociocognitively different.
(5) Decoding: The decoding process attempts to reconstruct the original data structure from the latent embeddings, ensuring that the latent representation captures the most import information in the data. During the decoding process, our methods simultaneously learns a rotation matrix that when multiplied by (3) generates low dimensional embeddings for each unit of analysis. These additional embeddings are injected into the decoding process to ensure that they align as closely as possible with the original inputs (separate social and cognitive networks), the encoded latent embeddings, and the decoded output.

The result of this process is an embedding for each unit of analysis in a two-dimensional space. These embeddings are obtained from the multiplication of the sociocognitive networks (2) and the rotation matrix (4). This allows us to interpret the dimensions of the embedding space in terms of the original features and the weights in the rotation matrix just as we do in ENA. Finally, we position the networks from (2) in the embedding space using the same optimisation routine used by the ENA algorithm, meaning that we can use the position of the network nodes and the midpoints of their edges to interpret the dimensions of the space, just as we do in ENA.

To more clearly demonstrate how to interpret NENA networks and their resulting embedding space, we show a simple example below (see Figure 2). The figure shows the multiplication of a set of simple social networks in adjacency matrix form (a) and a set of simple ENA networks in co-occurrence vector form (b) to create a sociocognitive NENA network (c). In the figure, (d) represents a simple rotation matrix generated from the GNN autoencoding process. NENA produces NENA scores (e) by multiplying a matrix like (c) with a rotation matrix like (d).

Regarding interpretation, first note that the diagonal of the social network matrix is set to zero, indicating that speakers do not get "credit" for appearing in their own windows—i.e., no self connections. This implies that they do not get credit in matrix (c) for any of their own ENA co-occurrences in (b) when doing the multiplication. They do, however, get credit in matrix (c) for the connections that those in their social network make. For example, speaker 1 will get credit for speaker 2's AB connection because speaker 2 is in speaker 1's social network. The strength of this credit is proportional to the strength of their social interaction.

The effect described above has two important implications. First, unlike ENA networks, NENA networks do not reflect the connections between codes made by the speaker and the teammates they communicated with. Instead, they represent the connections between codes that the speaker's social network tended to make. In other words, NENA networks show the kinds of cognitive connections that the people a certain person interacts with tend to make. *NENA networks are a representation of the company a person keeps and what they discuss.*

Second, unlike the dimensions of the ENA embedding space, the dimensions of the NENA embedding space do not distinguish units of analysis in terms of the relative frequency of the connections between codes that were made. Instead, they distinguish between units of analysis in terms of the relative frequency of connections that their social network tended to make. The more a unit's social network makes a certain connection between codes and the more the unit interacts with their social network, the greater the impact of component of the NENA dimension. In this simple example, we could interpret the first dimension in terms of individuals whose social networks made connections to AB vs BC. Because speaker 1 frequently interacts with people who make connection AB, their NENA score will be further to the right on the first dimension.

In the following subsections, we describe the NENA method in more detail.

3.4.1 *Input Layer.*

The algorithm begins with a set of nodes, each representing a unit of analysis, and weighted edges that indicate the frequency of interactions between these units within the same team, which represent the social interaction in the network. Each node is assigned features based on their normalized co-occurrence counts between codes, which are derived from ENA. These inputs follow the ENA model specifications, with edges identified using a moving window size of five. The GNN inputs are processed through multiple graph convolutional layers, and graph attention layers, to

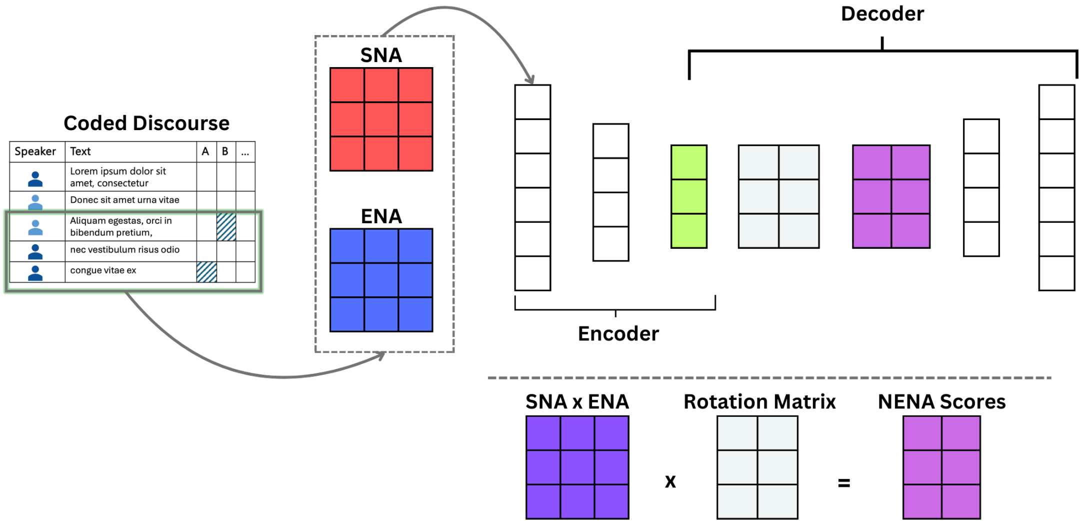


Figure 1: NENA architecture

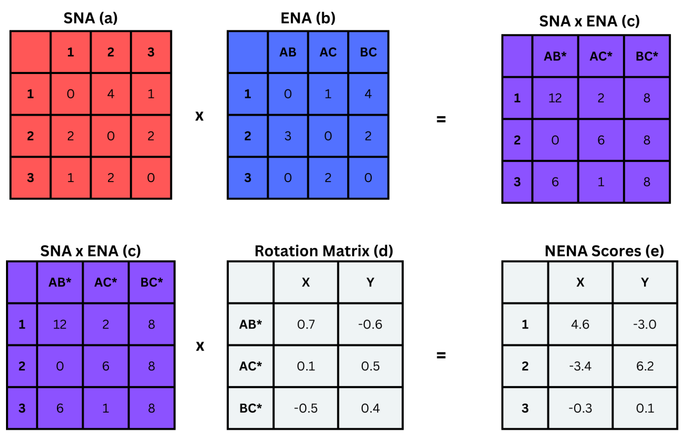


Figure 2: NENA Example

progressively refine the feature representation of each node within the overall graph.

3.4.2 GNN Autoencoder. In the new NENA structure, we incorporate a GNN autoencoder to effectively capture both cognitive and social dimensions in CPS. The autoencoder comprises two main components: an encoder and a decoder.

The encoder uses layers of GCNConv and GATConv to transform the input sociocognitive features into a latent space. The initial layers, GCNConv, perform convolution operations on the graph, aggregating information from neighboring nodes to generate higher-level feature representations. These are followed by GATConv layers, which use attention mechanisms to weigh the importance of each neighbor, further refining the node features based on their contextual relevance. This layered approach allows the model to learn intricate patterns and relationships within the data, resulting in latent embeddings that capture the essential characteristics of the nodes.

3.4.3 Rotation. Instead of directly using the GNN output embeddings, the NENA scores matrix, denoted as F, is used to determine the positions of embeddings for visualization. This matrix is obtained by multiplying the original sociocognitive matrix X with a learned rotation matrix E, where E is derived by passing the latent embeddings through a fully connected layer. This transformation ensures that the visualization remains consistent with the learned embeddings while preserving the interpretability and visualization strengths of ENA.

Afterward, the NENA scores matrix is passed through additional fully connected layers, initiating the decoding process. The decoder reconstructs the original features from F using further GCNConv and GATConv layers. These layers reverse the encoding process, progressively transforming the latent representations back into the original feature space, ensuring that the learned embeddings accurately reflect the structure of the underlying data.

3.4.4 Orthogonal Loss. The orthogonal loss enforces orthogonality constraints on the node embeddings from the GNN output. It is calculated as the sum of squares of the cosine similarities between all pairs of feature vectors. The orthogonal loss is defined as:

$$L_{\text{ortho}} = \frac{\sum_{i=1}^{n}\sum_{j=i+1}^{n}\left(\cos(\mathbf{x}_i,\mathbf{x}_j)\right)^2}{\frac{n(n-1)}{2}} \tag{1}$$

where:

- L_{ortho} is the orthogonal loss.
- $\mathbf{x}_i$ and $\mathbf{x}_j$ are the feature vectors.
- n is the number of features.
- $\cos(\mathbf{x}_i,\mathbf{x}_j)$ is the cosine similarity between feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$.

This regularization ensures that the feature vectors remain as orthogonal as possible, aligning with ENA's objective to represent distinct and non-redundant aspects of the data.

3.4.5 Graph Contrastive Loss. The core loss function in our architecture is the graph contrastive loss, which aims to bring embeddings of similar nodes closer together while pushing embeddings of dissimilar nodes farther apart in the embedding space. This loss function includes both positive and negative components.

Positive Loss. The positive loss is computed for nodes connected by social network edges in the graph, representing existing interactions between units of analysis. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be the reconstructed feature vectors for nodes i and j, respectively, and $\mathbf{A}$ be the adjacency matrix representing the graph's edge weights. The positive loss, L_{pos}, is defined as:

$$L_{\text{pos}} = \frac{1}{\sum_{(i,j)\in\mathcal{E}} a_{ij}} \sum_{(i,j)\in\mathcal{E}} a_{ij}\|\mathbf{x}_i-\mathbf{x}_j\|^2 \tag{2}$$

where $\mathcal{E}$ denotes the set of edges in the graph, a_{ij} is the weight of the edge between nodes i and j, and $\|\cdot\|^2$ represents the squared Euclidean distance.

Negative Loss. To separate embeddings of non-interacting nodes, we introduce negative sampling. We randomly sample pairs of nodes that are not connected by edges—i.e, not connected in the social network. Let $\mathbf{x}_k$ and $\mathbf{x}_l$ be the feature vectors of two such randomly selected nodes. The negative loss, L_{neg}, is calculated as:

$$L_{\text{neg}} = \frac{1}{N_{\text{neg}}} \sum_{(k,l)\notin\mathcal{E}} \max(0, m-\|\mathbf{x}_k-\mathbf{x}_l\|^2) \tag{3}$$

where m is a margin parameter ensuring that the negative pairs are sufficiently separated, and N_{neg} is the total number of negative samples.

Total Contrastive Loss. The total graph contrastive loss, $L_{\text{contrastive}}$, combines both the positive and negative losses:

$$L_{\text{contrastive}} = L_{\text{pos}} + L_{\text{neg}} \tag{4}$$

3.4.6 Isometry Loss Function. To enhance the isometric properties of the embedding space in the new NENA model, we introduced a specialized loss function aimed at preserving the pairwise distances between the original and transformed data points. This loss function operates by first calculating the pairwise Euclidean distances for both the embeddings (NENA points) and the original points. By focusing on the relationships between data points, the function ensures that the structural integrity of the original space is maintained in the transformed space. The process involves measuring the Pearson correlation between the distance matrices of the transformed and original points. This correlation quantifies the degree to which the pairwise distances are preserved after transformation. The loss function then aims to minimize the negative correlation, the loss function effectively maximizes the preservation of pairwise distances, thereby enhancing the isometric properties of the embedding space. This approach ensures that the transformation process retains the geometric relationships critical for accurate and interpretable visualizations in the NENA model.

3.5 Comparative Analysis

In summary, both ENA and the new version of NENA generate two-dimensional embeddings for each unit of analysis in the dataset. These embeddings, along with their corresponding networks, are visualized and analyzed both qualitatively and statistically. The main difference between ENA and NENA lies in the networks visualised and the procedure used to generate the rotation matrix used for embedding. ENA networks represent connections that an individual makes among codes in conversation with their teammates. NENA networks represent connections among codes that an individual's social network makes. When looking for differences between two subpopulations in the data, ENA uses means rotation and SVD to perform an isometric rotation to orthogonal dimensions. NENA uses a rotation matrix learned from the GNN autoencoder that aims to provide an othorgonal and isometric rotation

We compared the performance of NENA to ENA in terms of (a) the interpretability of the embedding spaces; (b) the goodness of fit between network centroids and embeddings; (c) isometry; (d) orthogonality; (e) the ability to statistically distinguish between two subpopulations in the data ; and (f) the ability to distinguish between individual units of analysis in the data.

3.5.1 Embedding Space Interpretation. For the TADMUS dataset, we provided qualitative interpretations of each dimension of the ENA and NENA embedding spaces by analyzing the nodes and connections at the extremes of each dimension, as these reflect the most influential co-occurrence variables. To quantitatively compare the dimensional interpretations, we computed the correlation between the rotation matrices of each model. A high magnitude correlation indicates that both embedding spaces differentiate units of analysis in similar ways, suggesting comparable interpretations.

3.5.2 Goodness of Fit. The goodness of fit for the ENA and NENA models was assessed by calculating the correlation between the network centroids for each unit of analysis and their corresponding embeddings. A high correlation signifies a reliable interpretive relationship between the placement of the network nodes/edges and the embedding space dimensions.

3.5.3 Isometry and Orthogonality. Isometric rotations preserve the pairwise distances between observations before and after rotation. To compare the isometry of the ENA and NENA models, we computed the Euclidean distances between each pair of observations before and after the rotation and then calculated the correlation between these two sets of distances for each model. High correlation indicates high isometry. To evaluate orthogonality, we calculated the correlation between the embedding values on the first and second dimensions of the space. A correlation of zero signifies complete orthogonality.

3.5.4 Regression Analysis and Effect Size Bootstrap. To assess the predictive power of the ENA and NENA models, we conducted mixed-effects regressions on the TADMUS data, comparing embedding values of known subgroups. Specifically, we regressed the embedding values on categorical predictors for condition (experimental vs. control), scenario, role, and included a random effect for individuals nested within teams. The key coefficient of interest was that of the condition variable, indicating the difference between experimental and control conditions on each dimension.

For statistical comparison between the models, we bootstrapped the data, reran the regression models on each bootstrap sample, measured the significance of the coefficients of interest, and converted them into effect sizes as measured by Cohen's d. Using 1000 bootstrap samples, we generated a distribution of effect sizes for each model. Additionally, we calculated the proportion of statistically significant coefficients in each model.

3.5.5 Distinguishing Individuals. If NENA is a useful analysis approach, it should be able to provide information that ENA does not. To this end we identified two individuals whose ENA scores were very close together but their NENA scores were much farther apart in their respective embedding spaces. In other words, we identified two individuals that ENA could not meaningfully distinguish, but that NENA could. As an example of the utility of NENA, we visually compare the differences between these individual indicated by the two models.

4 Results

4.1 Embedding space interpretation

Figure 3 shows the embedding spaces for the NENA model (left) and ENA model (right). Here, we show the network subtraction between the mean networks of the control condition (red) and the experimental condition (blue) in each plot. The nodes (and connections) positioned at the extremes of the dimensions distinguish observations the most, and thus can be used to interpret the dimensions of the embedding spaces.

For the NENA model, the first dimension (X) distinguishes *those whose social networks made stronger connections* to TrackBehavior on the left versus those whose social networks made stronger connections among SeekingInformation, StatusUpdate, DetectIdentify and Recommendation on the right. The second dimension (Y) distinguishes those whose social networks made stronger connections among DetectIdentify,Recommendation, DeterrentOrders and DefensiveOrders at the top versus those whose social networks made stronger connections to StatusUpdate and TrackBehavior at the bottom. For the ENA model, the first dimension distinguishes *individuals who made stronger connections* to SeekingInformation on the left versus those who made stronger connections to Recommendation, DeterrentOrders and StatusUpdate on the right.[1]

In the NENA model, several nodes maintain high importance at the extremes of the dimensions, similar to the ENA model. These include DeterrentOrders, StatusUpdate, DetectIdentify, Recommendation and SeekingInformation. However, the importance of other nodes varies between the two models. For instance, TrackBehavior shows high importance in the NENA model but lower importance in the ENA model, being closer to to the origin. Both models are able to clearly distinguish between participants in the experimental and control conditions—NENA on the Y dimension and ENA on the X dimension. The correlation between the rotation matrices for these spaces is **0.063**[2]

4.2 Goodness of Fit

Regarding the (GOF) values for each model, which show the correlation between network centroids and embeddings on the X and Y dimensions, the NENA model has very high correlations on both dimensions (**0.9871** and **0.9728**), indicating its robustness in preserving the relationships between node positions and embeddings. The traditional ENA model also shows strong correlations (**0.9595** on X and **0.9528** on Y).

4.3 Isometry and Orthogonality

The isometry scores reflects the degree to which transformations preserve distances between points. The NENA model has an isometry score of **0.9999**, which is very close to perfect. This slightly lower score aligns with the expectation that the GNN-enhanced NENA model, being nonlinear, would still approximate an isometric transformation effectively. In contrast, the ENA achieved a perfect isometry score of **1.00**. This is consistent with the linear nature of these models, which inherently maintain isometric transformations. These results reinforce the robustness of the NENA model, as it achieves near-perfect isometry despite its nonlinear approach, while the linear ENA models perform as expected. The correlation between the embeddings on the first and second dimensions was **0.002** for the NENA model and less than **0.001** for the standalone ENA model, suggesting that both models maintain orthogonality between the first and second dimensions, which is crucial for interpretability.

4.4 Effect-Size Bootstrap

Table 4 presents the results of the bootstrap analysis comparing the experimental conditions for both models. It reports the mean effect size (Cohen's d) and the 95% confidence interval for both dimensions of the two embedding spaces (ENA with mean rotation in green; NENA in blue). For the X axis, the average effect size

[1]Note that because the ENA model used a means rotation, which forces the average Y axis values to 0, interpreting the Y dimension is not needed.

[2]This value was calculated by taking the absolute values of the rotation matrix entries for each model and calculating the correlation between them. We take the absolute value because the orientation of any rotation matrix dimension is arbitrary; thus we are interested in the similarity of the magnitude of the entries in the rotation matrices when comparing them between models.

(a) NENA **(b) ENA**

Figure 3: Network subtractions between individuals in the experimental condition (blue) and control condition (red) for the NENA model and the ENA models.

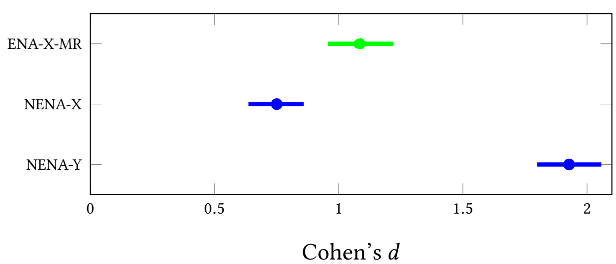


Figure 4: Bootstrap Confidence Intervals on Cohen's *d*.

for ENA with means rotation is $\bar{d}$ = 1.084, 95% CI [0.958, 1.218], with 99.7% of the associated p values being significant, indicating a strong ability to distinguish between conditions. The average effect size for NENA on the X axis is $\bar{d}$ = 0.751, 95% CI [0.636, 0.859], with 68% of the associated p values being significant. The largest average effect size is for the NENA Y axis: $\bar{d}$ = 1.928, 95% CI [1.799, 2.057], with 100% of the associated p values being significant.

4.5 Distinguishing Individuals

Figure 5 shows the network subtractions for two individuals who were found to be very close—i.e., similar—in ENA space, but more distinct in NENA space. Figure 5 (b) shows the networks and ENA scores for two individuals. Notice that the ENA scores (lower right) are overlapping in the space. Viewing the network subtraction we see that there were some differences in the kinds of connections between codes that they made. However, these differences are small, as indicated be the relatively thin and faint line weights. Connections with the thinnest/faintest line weights are among DetectIdentify, StatusUpdate, Recommendation, and DetterentOrders, indicating that the two individuals—one a commanding officer (CO) and one a supporting officer (IDS)—made these connections with similar frequency.

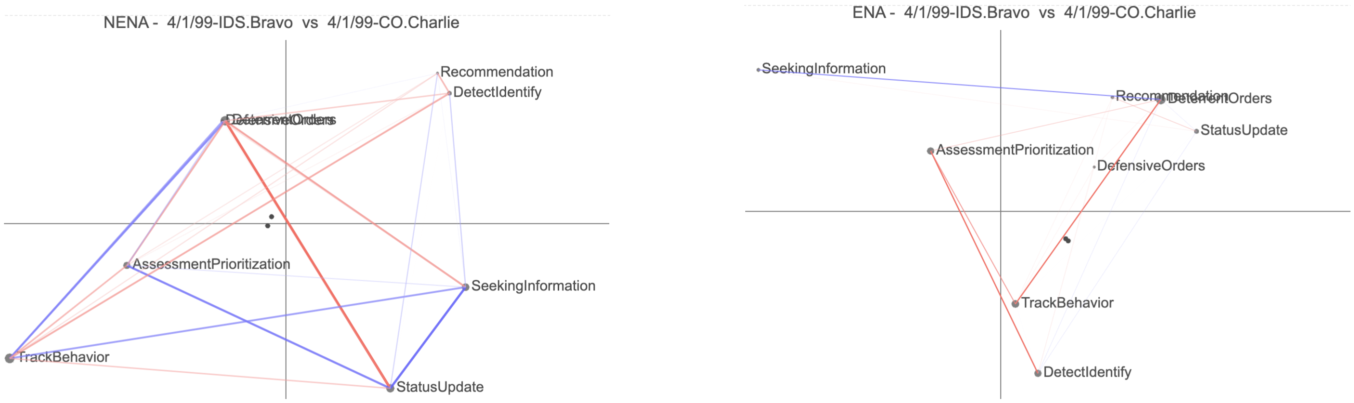
Figure 5 (a) shows the same individuals in NENA space. Here, we see that their NENA scores are non-overlapping and distinct in the space (middle-left). Recall that NENA networks show the connections that an individual's social network tends to make, not the connections that they make. The network subtraction indicates that the individual in a supporting role (IDS in blue) interacted with teammates who made more connections between: TrackBehavior, DefensiveOrders, and DeterrentOrders; TrackBehavior and SeekingInformation; and AssessmentPrioritization and StatusUpdate. Such connections are characteristic of the behaviour of *commanding officers* (see [23]). In other words, this individual tended to interact more with commanding officers.

In contrast, the network subtraction indicates that individual in the commanding role (CO in red) interacted with teammates who made more connections between: TrackBehavior and StatusUpdate; TrackBehavior, Recommendation, and DetectIdentify; DefensiveOrders, DeterrentOrders, StatusUpdate and SeekingInformation. Such connections are characteristic of connections that *supporting team members* tend to make. In other words, this individual tended to interact more with supporting team members.

These differences make sense in the context of this data. We would expect that supporting officers (e.g., IDS) would interact with commanding officers as their role is to pass information to commanding officers so that they can make effective decisions. Similarly, we would expect that commanding officers would interact more with supporting team members, as their objective is to gather information and make decisions. The NENA space and network subtractions highlight this expected behaviour.

5 Discussion and Conclusions

CPS is defined as an inherently sociocognitive phenomena [8]. Despite this, extant learning analytic techniques tend to focus on either the social of cognitive aspects without explicitly considering



(a) NENA **(b) ENA**

Figure 5: Network subtractions between 4/1/99-IDS.Bravo (blue) and 4/1/99-CO.Charlie (red) for the NENA model and ENA models.

their interaction. Prior work developed NENA, which used a combination of deep learning methods to simultaneously model the social and cognitive aspects of CPS [6]; however, the method had several limitations. The model was complex and sensitive to various hyperparameters, making it harder to implement and less generalisable; the method occluded social information in the interpretation of the final outputs; and it did not preserve the isometry of the original data points, which is critical for meaningful statically tests with distance-based measures. The refined version of NENA presented here addresses these limitations by (a) introducing a simplified autoencoder deep learning architecture; (b) using a combination of social and epistemic networks as input to preserve interpretability in terms of social and cognitive factors; and (c) introducing an isometry loss function into the architecture to ensure downstream statistical tests are meaningful.

We found that NENA is able to achieve high performance on criteria we would expect from a network analytic technique in the context of learning analytics: interpretability, goodness of fit, orthogonality and isometry; and discriminatory power. We also demonstrated that this method was comparable in performance to a more traditional learning analytic technique—ENA—while providing information that ENA did not. The results suggest that NENA could be a useful method for exploring the cognitive interactions of a given individual's social network and thus the influences their network exerts upon them. This is in contrast to ENA, which is more suitable for exploring the cognitive interactions of an individual in the context of their social network.

Our study has several limitations. Most obviously, we only evaluated NENA on one CPS dataset. Investigations of more datasets are needed to better understand the utility and behaviour of NENA and when it might be appropriate to use over a simpler technique like ENA. In our future work, we plan to explore this question not only with other real CPS datasets, but datasets of simulated collaborative discourse [21]. It might also be that a deep learning architecture is more complex than needed for such a problem. For example, we might get similar results to those presented here by simply applying means rotation and SVD to the multiplied social and epistemic networks. Although, we did not present it here, our initial tests suggested that such an approach did not yield suitable goodness of fit. However, more work will need to explore this possibility

Despite these limitations, this work presents a novel approach to modelling the social and cognitive aspects CPS. Our results suggest that it is comparable to existing learning analytics techniques for CPS data, while providing distinct insights. The method is particularly useful for research questions centred on the behaviours of a given individual's social network and the influence of that network on the individual.

References

[1] Tasweer Ahmad, Lianwen Jin, Xin Zhang, Songxuan Lai, Guozhi Tang, and Luojun Lin. 2021. Graph convolutional neural network for human action recognition: A comprehensive survey. *IEEE Transactions on Artificial Intelligence* 2, 2 (2021), 128–145.

[2] Jessica Andrews-Todd and Carol M Forsyth. 2020. Exploring social and cognitive dimensions of collaborative problem solving in an open online simulation-based task. *Computers in human behavior* 104 (2020), 105759.

[3] Dale Bowman, Zachari Swiecki, Zhiqiang Cai, Yeyu Wang, Brendan Eagan, Jeff Linderoth, and David Williamson Shaffer. 2021. The mathematical foundations of epistemic network analysis. In *Advances in Quantitative Ethnography: Second International Conference, ICQE 2020, Malibu, CA, USA, February 1-3, 2021, Proceedings 2.* Springer, 91–105.

[4] Pierre Dillenbourg. 1999. What do you mean by collaborative learning? *Collaborative-learning: Cognitive and Computational Approaches.* (1999), 1–19.

[5] Cynthia D'Angelo, Golnaz Arastoopour, Naomi Chesler, and David Williamson Shaffer. 2011. Collaborating in a virtual engineering internship. (2011).

[6] Blinded for review. [n. d.].

[7] Dragan Gašević, Srećko Joksimović, Brendan R Eagan, and David Williamson Shaffer. 2019. SENS: Network analytics to combine social and cognitive perspectives of collaborative learning. *Computers in Human Behavior* 92 (2019), 562–577.

[8] Arthur C Graesser, Samuel Greiff, Matthias Stadler, and Keith T Shubeck. 2020. Collaboration in the 21st century: The theory, assessment, and teaching of collaborative problem solving. , 106134 pages.

[9] Caroline Haythornthwaite. 1996. Social network analysis: An approach and technique for the study of information exchange. *Library & information science research* 18, 4 (1996), 323–342.

[10] Joan H Johnston, Stephen M Fiore, Carol Paris, and CAP Smith. 2013. Application of cognitive load theory to develop a measure of team cognitive efficiency. *Military Psychology* 25, 3 (2013), 252–265.

[11] Dmitry Kobak and George C Linderman. 2021. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature biotechnology* 39, 2 (2021), 156–157.

[12] Roberto Martinez-Maldonado, Dragan Gašević, Vanessa Echeverria, Gloria Fernandez Nieto, Zachari Swiecki, and Simon Buckingham Shum. 2021. What do you mean by collaboration analytics? A conceptual model. *Journal of Learning Analytics* (2021).
[13] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).
[14] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. 2018. Adversarially regularized graph autoencoder for graph embedding. *arXiv preprint arXiv:1802.04407* (2018).
[15] Muhammad Asif Qureshi, Asadullah Khaskheli, Jawaid Ahmed Qureshi, Syed Ali Raza, and Sara Qamar Yousufi. 2023. Factors affecting students' learning performance through collaborative learning and engagement. *Interactive Learning Environments* 31, 4 (2023), 2371–2391.
[16] Florian Schwade. 2021. Social Collaboration Analytics Framework: A framework for providing business intelligence on collaboration in the digital workplace. *Decision Support Systems* 148 (2021), 113587.
[17] D Shaffer and A Ruis. 2017. Epistemic network analysis: A worked example of theory-based learning analytics. *Handbook of learning analytics* (2017).
[18] David Williamson Shaffer. 2018. Epistemic network analysis: Understanding learning by using big data for thick description. In *International handbook of the learning sciences*. Routledge, 520–531.
[19] David Williamson Shaffer, Wesley Collier, and Andrew R Ruis. 2016. A tutorial on epistemic network analysis: Analyzing the structure of connections in cognitive, social, and interaction data. *Journal of Learning Analytics* 3, 3 (2016), 9–45.
[20] David Williamson Shaffer, David Hatfield, Gina Navoa Svarovsky, Padraig Nash, Aran Nulty, Elizabeth Bagley, Ken Frank, André A Rupp, and Robert Mislevy. 2009. Epistemic network analysis: A prototype for 21st-century assessment of learning. *International Journal of Learning and Media* 1, 2 (2009), 33–53.
[21] Zachari Swiecki, Cody Marquart, and Brendan Eagan. 2022. Simulating Collaborative Discourse Data. In *Proceedings of the 15th International Conference on Computer-Supported Collaborative Learning-CSCL 2022, pp. 83-90.* International Society of the Learning Sciences.
[22] Zachari Swiecki, AR Ruis, and David Williamson Shaffer. 2019. Modeling and visualizing team performance using epistemic net-work analysis. In *Proceedings of the 7th annual GIFT users symposium.* 148–156.
[23] Zachari Swiecki, Andrew R Ruis, Cayley Farrell, and David Williamson Shaffer. 2020. Assessing individual contributions to collaborative problem solving: a network analysis approach. *Computers in Human Behavior* 104 (2020), 105876.
[24] Zachari Swiecki and David Williamson Shaffer. 2020. iSENS: an integrated approach to combining epistemic and social network analyses. In *Proceedings of the tenth international conference on learning analytics & knowledge*. 305–313.
[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[26] Xiao-Meng Zhang, Li Liang, Lin Liu, and Ming-Jing Tang. 2021. Graph neural networks and their current applications in bioinformatics. *Frontiers in genetics* 12 (2021), 690049.
[27] Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S Yu. 2022. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082* (2022).