Who Plays Which Role When? Communication Role Dynamics for Peer Recognition and Team Performance Prediction

Yifan Song[1], Wenxuan Wendy Shi[2], Brian P. Bailey[1], Tal August[1]
[1]University of Illinois Urbana-Champaign
[2]California Polytechnic State University, Pomona
{yifan33, bpbailey, taugust}@illinois.edu, wendyshi@cpp.edu

Abstract

Team roles offer an interpretable lens on collaboration, yet computational studies of roles often rely on domain-specific personas or data-driven clustering rather than theory-grounded taxonomies. We operationalize a taxonomy of eight communication roles grounded in education literature and annotate a corpus of 6,307 Slack messages from 55 students across 18 teams in a semester-long computer science course project. We evaluate whether LLMs can approximate expert labels, enabling scalable, taxonomy-driven role annotation. Using these role labels, we characterize role dynamics over teams' lifecycles, finding that different roles peak at different moments and that students enact a more diverse set of roles as projects progress. To evaluate the utility of our role constructs, we use them to predict peer recognition, outperforming lexical, conversational, and LLM-prompting baselines. To assess generalizability beyond the educational context, we apply the same role constructs to a public dataset (DeliData) to predict team performance improvement after deliberation, again exceeding prior performance.

1 Introduction

Effective teamwork depends not only on what a team produces, but also on how members communicate, such as introducing ideas, coordinating work, and managing disagreement. A useful lens for describing these interaction patterns is *team roles*. Behavioral and organizational science conceptualizes roles as complementary communicative functions enacted in interaction (e.g., an *initiator* who proposes new ideas, or an *arbitrator* who solves disagreements), rather than as fixed personality traits (Benne and Sheats, 1948; Meredith Belbin, 2011). This perspective naturally raises a core question for computational analysis of team communication: who enacts which roles, when, and how do these patterns relate to collaboration quality?

Prior work has modeled functional roles for meeting participants from simple speech features (Banerjee and Rudnicky, 2006), or used behavioral patterns (e.g., turn-taking) to predict "latent" roles and team outcomes (Yang et al., 2015). Recently, large language models (LLMs) have renewed interest in roles through human-agent and agent-agent collaboration, where agents are assigned explicit social roles in games or simulated organizations (Lan et al., 2024; Li et al., 2025a). However, roles are often operationalized as domain-specific personas or functions (e.g., *developer* vs. *manager*), rather than grounded to theory-driven communicative functions. In addition, most prior work studies roles in large-scale (e.g., Wikipedia), controlled (e.g., crowdsourced), or synthetic (e.g., agent simulation) contexts (Maki et al., 2017; Litman et al., 2016; Lu et al., 2024). While these contexts offer scale and control, they differ from the close-knit, long-lived teams common in educational, research, and organizational settings (Mathieu et al., 2017), and often miss how real teams evolve over time.

In this paper, we ground team roles in an authentic longitudinal setting: an in-person, semester-long computer science course project in which teams relied on Slack for day-to-day coordination. We collect a dataset of 6,307 messages from 55 students across 18 teams over eight project deliverable windows. We operationalize a role taxonomy grounded in the education literature (Nestsiarovich and Pons, 2020) and develop an expert annotation protocol for labeling the eight roles based on the taxonomy. We then evaluate an LLM-as-annotator setup as a scalable approximation to expert annotation, with the best model yielding good overall agreement with expert labels.

Using these role labels, we provide descriptive analyses of role dynamics—the longitudinal evolution of an individual's communicative functions—including role prevalence and trajectories over the project lifecycle. For example, work-related roles

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 41615–41628

like *explorer* and *facilitator* peak during the intensive implementation phase, while *gatekeeper* increases substantially near the end of the project during the busiest and most stressful period. We also observe a progressive increase in the number of unique roles an individual enacts simultaneously as project complexity grows.

To evaluate the usefulness of these role constructs, we first predict peer recognition, an individual-level performance measure derived by normalizing peer-evaluation scores. We show that role-based features improve predictive performance over lexical, conversational, and zero-shot LLM baselines, yielding about 5–10% improvement. To further validate generalizability beyond the educational context, we use the same LLM-annotated role constructs to predict team performance gain in a public dataset (DeliData, Karadzhov et al., 2023) of group deliberation dialogues. Combining our role features with conversational statistics outperforms all baselines reported in the original work.

Our contributions include:

- We show that a theory-grounded role taxonomy can be used to reliably annotate roles in a dataset of student team conversations, and that LLMs can approximate expert annotation.
- We characterize role prevalence and trajectories across a semester-long project, clarifying how role dynamics change in response to specific project demands.
- We demonstrate that our role constructs improve downstream-task performance on peer recognition and team performance prediction across two different datasets.

2 Related Work

Prior studies have linked linguistic and pragmatic features (e.g., politeness, toxicity) to task success and conversational failure (Reitter and Moore, 2007; Zhang et al., 2018). Such interpretable features can also help predict and explain team-level performance, such as team viability (Cao et al., 2021). Beyond conversational features, prior work has also used topological and structural features of communication networks (Ghawi et al., 2021) or static personal traits (e.g., grades and MBTI, Omar et al., 2011) to predict team performance.

Within this line of work, some researchers have specifically studied how roles affect communication and teamwork. Banerjee and Rudnicky (2006) predicted functional roles during meetings (e.g., *leader*, *scribe*) from lexical and participation cues, while Dong et al. (2013) identified functional roles from non-linguistic cues (e.g., physical fidgeting). Yang et al. (2015) and Maki et al. (2017) leveraged weakly supervised approaches to induce latent role representations from turn-taking behaviors and stylistic markers to predict team outcomes in MOOCs and online communities. Recent LLM-based multi-agent frameworks go further by assigning social roles to agents. Studies show that LLMs can simulate diverse character profiles (Lu et al., 2024) and exhibit emergent social behaviors, such as leadership, confrontation, and persuasion, in multi-agent environments (Lan et al., 2024; Li et al., 2025a). However, these studies approach role modeling primarily in short sessions under synthetic laboratory or simulated environments, whereas our work instead detects roles from longitudinal, real-world team conversations.

Unlike most prior role-modeling research, we ground our role definitions in organizational and behavioral science theories that conceptualize roles as communicative functions that support task and socio-emotional processes (Benne and Sheats, 1948; Meredith Belbin, 2011; Mathieu et al., 2015). In engineering education, researchers have also found that appropriate role divisions benefit team-based learning from team formation to assessment (Jahanbakhsh et al., 2017; Aranzabal et al., 2022). To build a role taxonomy specifically for this domain, Nestsiarovich and Pons (2020) used observation-driven studies of engineering classroom teams using interaction diagrams. Our work builds on this taxonomy by operationalizing their roles in student Slack conversations.

3 Dataset

We collect our dataset from a semester-long team project in an upper-level computer science course at a large public research university. The course topic was user interface design, and the team project accounted for 45% of the grade. Students worked in teams of four to six, and most students were upper-year undergraduates or early graduate students.

The course was taught in person, and teams both met in person and communicated online. Each team was provided a private channel within the course Slack workspace as a default communication space for brainstorming, planning, and task

Statistic	Value
Students analyzed / consented / enrolled	55 / 94 / 186
Teams represented	18
Student-deliverable instances	424
Total messages	6,307
Total words	93,474
Median messages per instance	9
Median words per instance	116

Table 1: Dataset summary. "Students analyzed" refers to consenting students whose teams actively used the course Slack channel throughout the semester.

updates. Teams could also coordinate using other communication channels. The project included eight graded deliverables, indexed as D1–D8: planning & proposal (D1–D2), user research (D3), low-fidelity prototyping & evaluation (D4–D5), and functional prototype implementation & evaluation (D6–D8). Each deliverable window lasted one to two weeks.

At the end of the semester, 94 out of 186 enrolled students consented to share their course data for research. Our analysis focuses on consenting students whose teams actively used the course Slack channel throughout the semester (having more than 100 messages in total). We aggregate Slack messages by author and deliverable window, yielding 424 student-deliverable instances. Each instance consists of the set of messages authored by one consenting student during one deliverable window. Because consent was obtained at the individual level rather than the team level, not all members of a given team are necessarily represented in the dataset. To avoid assumptions about unobserved teammate behavior, all modeling and analyses are conducted at the individual (student-deliverable) level. This study was approved by the Institutional Review Board of the authors' university.[1]

4 Roles in Team Communication

We adopt a theory-driven role taxonomy from Nestsiarovich and Pons (2020), which was developed via in-situ observation of engineering project teams and characterizes roles as communicative functions (e.g., initiating work, facilitating coordination, regulating participation). Compared to other popular role taxonomies designed for broader organizational settings (e.g., Meredith Belbin, 2011; Mathieu et al., 2015), this taxonomy from education literature aligns closely with our setting of task-oriented collaboration in student project teams.

[1]Raw student messages cannot be released due to privacy constraints. All data were anonymized during analysis. We share prompts and modeling details in the appendix.

Role	Shorthand description
Initiator	Initiate process
Explorer	Ask questions
Information Provider	Provide detailed information
Facilitator	Summarize / control discussion
Arbitrator	Solve disagreement
Representative	Express / answer for team
Gatekeeper	Fill gaps / invite others
Connector	Connect people/resources
Passive Collector	Collect information
Outsider	Stay outside

Table 2: Team role taxonomy from Nestsiarovich and Pons (2020) with shorthand descriptions, complete communication patterns for each role can be found in Appendix Table 8. *Passive collector* and *outsider* are excluded from annotation and modeling.

Table 2 summarizes the ten roles in the original taxonomy. In our study, we focus on the eight constructive roles and exclude *passive collector* and *outsider*. This choice reflects the limits of the partial observation in our Slack dataset: it is difficult to infer the negative roles of low engagement from individual messages alone. Silence or low messaging can be captured more directly via explicit conversational volume statistics (e.g., message/word counts) that we model separately (more details in Section 5.2), but these signals do not necessarily mean that a student is an *outsider* or *passive collector*, since they may be heavily engaged in off-platform work (e.g., programming) or other communication channels (e.g., video meetings). We further adapt role cues to Slack-style coordination (e.g., only project-related messages count for *explorer* or *information provider*, rather than asking simple scheduling questions or providing availability). The full taxonomy with typical communication patterns, which serve as annotation guidelines, can be found in Appendix Table 8.

4.1 Human Role Annotation

Using our adapted taxonomy, we annotate roles on each student–deliverable instance. Because roles are not mutually exclusive, we treat role labeling as a multi-label task: for each instance, annotators make an independent binary decision for each of the eight roles.

Two annotators with expertise in collaborative learning research—both are authors of this paper and former teaching staff for the course under

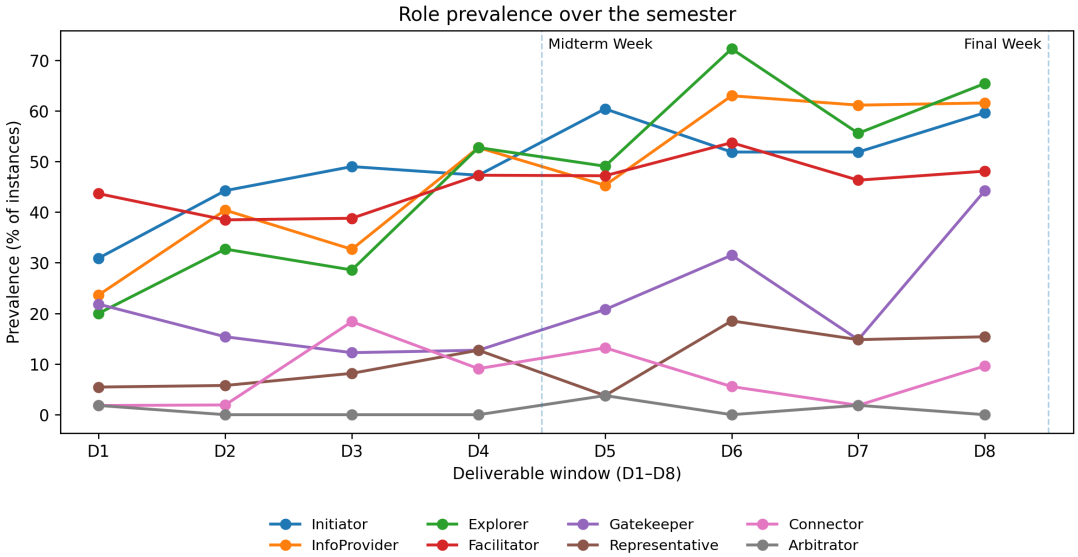

Figure 1: Role prevalence across deliverable windows. Vertical dashed lines mark midterm week (between D4 and D5) and final week (after D8). Peaks align with phase-specific demands: *connector* rises during user research (D3), work/coordination roles (*explorer*, *information provider*, *facilitator*) increase during first functional prototyping (D6), and *gatekeeper* is highest during final evaluation (D8).

Role	Support	Student Coverage
Initiator	209 (49.3%)	53 (96%)
Information Provider	202 (47.6%)	52 (95%)
Explorer	200 (47.2%)	50 (91%)
Facilitator	193 (45.5%)	48 (87%)
Gatekeeper	92 (21.7%)	44 (80%)
Representative	45 (10.6%)	31 (56%)
Connector	32 (7.5%)	23 (42%)
Arbitrator	4 (0.9%)	4 (7%)

Table 3: Role distribution in the expert-labeled student-deliverable instances (N=424), reporting how often a role is present across instances; student coverage represents how many students have enacted a role at least once throughout the semester (N=55).

study—labeled the dataset. We used a staged procedure to calibrate annotation and assess reliability. First, both annotators independently labeled an initial 10% of the instances, discussed disagreements, and refined the guidelines. Next, annotators independently labeled an additional 20% of the instances and confirmed reliability (Krippendorff's $\alpha = 0.867$). Disagreements in the double-labeled portion (30% total) were adjudicated to produce a single reference label, and the remaining instances were single-annotated.

Table 3 summarizes the distribution of expert-annotated roles. Four work and coordination roles: *initiator*, *explorer*, *information provider*, and *facilitator* appear in roughly half of instances and most students have enacted such roles at least once, whereas *gatekeeper*, *representative*, and *connector* are comparatively infrequent. *Arbitrator* is rarely observed in these Slack logs, suggesting that explicit conflict mediation is either uncommon in this setting or more likely to occur off-platform.

4.2 Role Dynamics

To connect our role framework to concrete team behaviors, we analyze role prevalence and trajectories over deliverable windows and describe our observations below.

Roles shift with project phases. Figure 1 shows how role prevalence varies across the project timeline with several roles exhibiting phase-aligned peaks. *Connector* spikes during the user research phase (D3), consistent with recruiting participants and reaching out to instructors/TAs or external resources when students first practice user research skills. Roles tied to technical clarification and coordination peak during the first functional prototyping window (D6): *explorer*, *information provider*, and *facilitator* increase, aligning with the first intensive implementation and debugging cycle. *Gatekeeper* rises sharply in the final evaluation window (D8), likely due to the heightened need to keep communication channels open during the busiest period of a semester—preparing finals and deadlines for all the courses. *Initiator* also peaks around D5 (adjacent to midterm) and again near D8 (adjacent

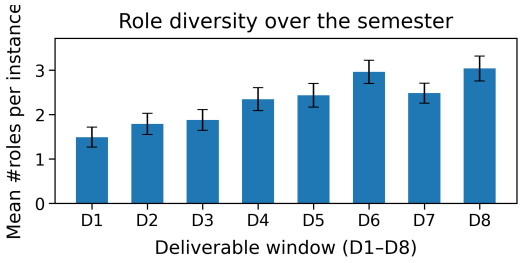

Figure 2: Mean number of roles enacted per student increases across project phases.

Role	F1	Krippendorff's α
Initiator	0.916	0.830
Explorer	0.924	0.858
Info Provider	0.888	0.787
Facilitator	0.867	0.769
Arbitrator	0.750	0.748
Representative	0.485	0.414
Gatekeeper	0.712	0.636
Connector	0.755	0.738
Micro average	0.856	0.799
Macro average	0.787	0.723

Table 4: Best LLM performance on role labeling (GPT-5.1) against expert labels (N=424). We report F1 and Krippendorff's α.

to final), possibly suggesting increased "driving" behaviors when time pressure is salient.

Role diversity increases over time. Within a single deliverable window, students typically enact multiple roles (mean = 2.3 roles per student-deliverable instance), reflecting that roles behave as situational communicative functions rather than mutually exclusive personal identities. Consistent with increasing project complexity and coordination demands, the mean number of roles enacted per student instance rises across the semester (Figure 2), suggesting that students begin to take on more roles as they settle into their teams.

4.3 LLM as Annotator

4.3.1 Methods

Because manual role labeling does not easily scale to a large number of team interactions, we evaluate whether LLMs can replicate our expert role annotations, following recent evidence that codebook-aligned prompting can approximate human coding for complex social constructs (Jia et al., 2024). We prompt the model with: 1) our adapted role definitions, 2) the ordered and anonymized messages from a student-deliverable instance, and 3) instructions to output an 8-dimensional binary role vector together with brief reasoning and evidence for each predicted positive role. We use a zero-shot prompt since few-shot generally performed worse, likely due to the nuances in chat history. The full prompt template is included in Appendix B.1.

4.3.2 Results

We compare the model outputs against the expert labels on the same N = 424 instances using per-role F1 (treating expert annotations as gold-standard labels) and Krippendorff's α (treating expert annotations and LLM predictions as two coders). We experimented with multiple model families and variants, including proprietary models (GPT and Gemini) and open-source models (DeepSeek), as well as variations in model size (regular vs. mini) and inference mode (reasoning vs. non-reasoning). Table 4 reports results for the best-performing model variant, GPT-5.1 (`reasoning={"effort":"low"}`). The full configurations and results of cross-model comparison are reported in Appendix B.2 & B.3.

When averaged across all instances, the best-performing model achieves a micro F1 of 0.856 and a micro Krippendorff's α of 0.799, suggesting LLMs can be used as a scalable proxy for role annotation. When averaged at the role level, the macro averages are 0.787 and 0.723, respectively. Following the rule-of-thumb thresholds in Krippendorff (2004), the four most common roles (*initiator*, *explorer*, *information provider*, and *facilitator*) yield good or excellent scores ranging from 0.77 to 0.86. The scores of less common roles (*arbitrator*, *gatekeeper*, and *connector*), ranging from 0.64 to 0.75, are still considered acceptable, especially in LLM labeling tasks with subjective or social role labels (August et al., 2020; Jia et al., 2024).

In contrast, *representative* is the hardest to annotate (F1 = 0.485, Krippendorff's α = 0.414), and is consistently challenging across different models, likely because "speaking for the team" often depends on conversational and social context beyond a single person's message history. Expert annotators, as former teaching staff, may leverage additional contextual knowledge about typical teamwork processes in this course setting. While *representative* only accounts for 10.6% of the total instances, the false positives frequently co-occur with strong *initiator* and *facilitator* signals, suggesting the model sometimes over-interprets generic coordination or information exchange as represen-

tational behavior. This error mode is consistent with prior annotation findings that agreement varies substantially across categories and is affected by frequency (Wegmann et al., 2024).

5 Task 1: Predicting Peer Recognition

While role prediction itself is valuable for determining how team members communicate, we are further interested in observing the downstream benefits of roles in predicting individual and team success. Here we explore the benefits of our role constructs in two downstream tasks. We first use roles to predict peer recognition in our dataset (Section 5), then predict team performance in an external public dataset (Section 6).

5.1 Task Setup

The first task aims to predict peer recognition from teammates, represented by peer evaluation ratings collected in the same course as our Slack chat dataset. After each deliverable, students completed a peer evaluation in which each student rated each teammate on a 5-point scale (*1 = below expectations, 3 = meet expectations, 5 = beyond expectations*) with optional comments. Course staff summarized received ratings into an *Adjustment Factor (AF)* for each student and deliverable window. For student i during deliverable window t,

$$\mathrm{AF}_{i,t} = \frac{\text{mean peer rating received by } i \text{ on } t}{\text{mean team rating on } t}.$$

An AF of 1.0 indicates that a student is rated on par with their team's mean for that deliverable window. Following prior work that predicts collaboration outcomes via meaningful binary splits rather than regressing a continuous score (Cao et al., 2021), we frame peer recognition prediction as a binary classification task over AF splits. Based on course curriculum, course staff flagged students based on thresholds of AF:

- $\mathrm{AF} < 0.95$ potential concerning performance;
- $0.95 \leq \mathrm{AF} < 1.05$ within the team's typical range;
- $\mathrm{AF} \geq 1.05$ potential outstanding performance.

Repeated occurrences with $\mathrm{AF} < 0.95$ were considered for penalty review and repeated $\mathrm{AF} \geq 1.05$ were considered for bonus review. These two thresholds, plus the average threshold of AF = 1.0, naturally form meaningful splits on AF. Table 5 summarizes class balance. We collected the peer evaluation data from consented students. We used AF as the outcome variable because it was a real metric used in the original course, and is normalized by the team average, which controls for team-level differences in rating.

Task split	Positive	Negative
Penalty ($\mathrm{AF} < 0.95$)	66 (15.6%)	358 (84.4%)
Bonus ($\mathrm{AF} \geq 1.05$)	109 (25.7%)	315 (74.3%)
Above Avg ($\mathrm{AF} \geq 1.0$)	218 (51.4%)	206 (48.6%)

Table 5: Class distribution of three binary classification tasks for peer recognition prediction (N=424).

5.2 Features

Bag-of-words (BoW) Each instance is represented as a BoW vector over n-grams (n = 1, 2, 3) as a lexical baseline (Hundhausen et al., 2023).

Conversational Features To construct conversational baselines, we extract features for each *student-deliverable instance* that are commonly used for group conversation prediction tasks from prior work.

- **Volume:** Message count and word count (Niculae and Danescu-Niculescu-Mizil, 2016).
- **Language:** Per-message polarity, subjectivity, toxicity, and readability and aggregate each score across messages using {min, max, mean, std} to capture a range of expression cues.[2] (Nguyen et al., 2016)
- **Disagreement:** Count of explicit disagreement markers (e.g., "*disagree*", "*but*"), drawn from the ARGUE corpus (Allen et al., 2014).
- **Interaction:** Count of direct teammate references (e.g., names and @mentions), as a proxy for social attention (Shibani et al., 2017).

Role Features (Ours) We represent each instance as an 8-dimensional binary vector indicating the presence of each role. We evaluate two versions of this feature set: 1) Roles (Human) as an upper bound using expert labels; and 2) Roles (LLM) using best-performing LLM labels.

Role + Conversation To test for complementarity between high-level roles and low-level cues (e.g., a student might be an *initiator* but also chat with negative sentiment), we concatenate the role vector with the conversation-feature vector.

[2]Used `TextBlob` to compute polarity and subjectivity, `Detoxify` to compute toxicity, and `textstat`'s Dale-Chall score to compute readability.

	Above Avg ($AF \geq 1.00$)	Penalty Risk ($AF < 0.95$)	Bonus Potential ($AF \geq 1.05$)
Baselines			
BoW	0.634 ± 0.046	0.665 ± 0.034	0.596 ± 0.042
Conversational Features	0.648 ± 0.029	0.640 ± 0.046	0.619 ± 0.034
Role features (ours)			
Roles only (Human)	$\mathbf{0.746 \pm 0.015}$	0.741 ± 0.062	0.679 ± 0.051
Roles only (LLM)	0.729 ± 0.027	0.730 ± 0.082	0.680 ± 0.078
Roles (Human) + Conversation	0.745 ± 0.012	$\mathbf{0.762 \pm 0.064}$	$\mathbf{0.687 \pm 0.037}$
Roles (LLM) + Conversation	0.723 ± 0.024	0.732 ± 0.066	0.685 ± 0.058
Zero-shot LLM			
Chat only	0.699	0.694	0.627
Chat + Roles	0.710	0.681	0.649

Table 6: ROC-AUC for predicting peer recognition. Supervised models report 5-fold student-grouped cross-validation mean $\pm$ std; zero-shot LLM baselines are single-pass evaluations.

5.3 Models

Supervised Classifier For feature sets that are already structured (BoW, conversation, role), we use Logistic Regression as the primary supervised classifier for its interpretability and stable performance on modest-sized datasets, following the approach in Cao et al. (2021). Implementation details are provided in Appendix C.1.

Zero-shot LLM Prompting Complementing the supervised classifier, we evaluate a zero-shot prompting approach to benchmark against pure LLM reasoning without explicit feature learning. We test two settings:

- **Chat-only Baseline:** Raw student-deliverable message sequence as input to predict probabilities for each binary AF split. This serves as a baseline without role operationalization.
- **Chat + Roles:** The model is provided with both message sequence and labeled role information. This benchmarks whether the LLM's internal reasoning can leverage role information better than our supervised classifier.

Both approaches use the best-performing LLM in role annotation (GPT-5.1). Detailed prompts are reported in Appendix C.2.

5.4 Evaluation

We report ROC-AUC as the primary metric, which is robust to class imbalance, following the same setting from Cao et al. (2021). Supervised models are evaluated with 5-fold cross-validation where folds are grouped by student identity (i.e., all instances from a student appear in exactly one test fold), preventing leakage across a student's multiple deliverable windows. We report mean $\pm$ std across folds. Zero-shot LLM baselines are evaluated once over the full dataset (no training), thus only one score is reported.

5.5 Results

Table 6 reports results from our different modeling approaches. Role-based representations provide substantially stronger predictive signal than lexical and conversational baselines across all three tasks. While BoW and conversational features yield modest performance (ROC-AUC $\approx$ 0.60–0.66), role features reach 0.746 / 0.741 / 0.679 (Above Avg / Penalty / Bonus) using expert role labels, and remain competitive when roles are produced by LLM annotation (0.729 / 0.730 / 0.680). This suggests that theory-grounded roles capture structure in team communication that is not well recovered by surface-level lexical variation or generic conversation statistics alone, and that LLM-based role annotation is a viable substitute when expert coding is unavailable.

Zero-shot prompting is better than other baselines but weaker than supervised role-based models. Prompting on student chat messages with or without roles yields similar results, suggesting that role features are better used as structured constructs, rather than integrating in end-to-end prompting.

Combining roles with conversational features yields the strongest performance for penalty and bonus prediction (0.762 and 0.687, respectively). In predicting above vs. below average, role features alone perform comparably, suggesting that role indicators already capture much of the signal needed to separate above- vs. below-average peer recogni-

tion, while conversational statistics help more in identifying more extreme situations.

Finally, task difficulty differs by split: predicting penalty tends to be easier than predicting bonus. A plausible explanation is that under-performance cues (e.g., lack of coordination or low messaging) are more directly reflected in chat behavior, whereas exceptional contributions may also occur outside Slack (e.g., implementation work).

6 Task 2: Predicting Team Performance

We selected a second task of predicting team-level performance using complete team dialogues, compared to the first task of predicting peer recognition at individual level. Our goal with this second task was two-fold: 1) to evaluate the generalizability of our role constructs beyond our dataset and educational context, and 2) to test our role constructs in a setting where team interaction was fully observed (i.e., all conversations were collected).

6.1 The DeliData Corpus

We evaluate on DeliData (Karadzhov et al., 2023), a public dataset of multi-party collaborative deliberation on the Wason card selection task (Wason, 1968). DeliData contains 500 group dialogue transcripts (2–5 members per group, 1,579 total participants, 14,003 utterances). Each dialogue is paired with objective correctness before and after discussion, enabling prediction of *conversational performance gain*—whether the group improves after deliberation. We frame performance gain as a binary classification task, reporting ROC-AUC under a leave-one-out cross-validation (LOOCV) setting, by following Section 7.1 of the original DeliData work.

6.2 Role Features from LLM Annotation

We apply the same role taxonomy (Section 4) and the same prompting setup using GPT-5.1 to infer participant roles from dialogue. Because DeliData provides the full dialogue transcript from all team members, we evaluate two annotation contexts:

- **Individual context:** label each participant using only their own utterances (analogous to our main dataset setting).
- **Team context:** label each participant while providing the full team transcript.

Since the downstream label of performance gain is defined at team level, we aggregate participant-level role labels into a team-level feature vector.

Feature Set	ROC-AUC
Reported by Karadzhov et al. (2023)	
(1) Deliberation Annotation	0.53
(2) Interaction Features	0.49
(3) Participation Dynamics	0.61
(4) Conversational Statistics	0.65
(1) + (2) + (3) + (4)	0.70
Ours	
(5) Zero-shot LLM Baseline	0.68
(6) Roles (Individual Context)	0.66
(7) Roles (Team Context)	0.69
(6) + (4)	0.72
(7) + (4)	**0.74**

Table 7: Predicting conversational performance gain on DeliData. (1)–(4) and their combination are reported from Table 5 of Karadzhov et al. (2023). (6)–(7) use roles annotated by LLM. Rows combining roles with conversational statistics use our re-implementation.

For a team with n members, let $y_{i,r} \in \{0, 1\}$ denote whether participant i exhibits role r. We compute the proportion of role r for team t:

$$p_{t,r} = \frac{1}{n} \sum_{i \in t} y_{i,r},$$

This results in an 1×8 vector for the eight role proportions, each with a value within $[0, 1]$. This normalization makes role features comparable across different team sizes.

6.3 Results

Table 7 compares our role-based features to representative baselines reported in Karadzhov et al. (2023). Using role proportions alone, both variants are competitive: roles inferred from team context outperform those inferred from individual context (0.69 vs. 0.66), suggesting that access to the full team transcript helps the model infer roles that depend on interactional context. This confirms that our role constructs provide additional benefits when data collection covers the whole team.

Motivated by the complementarity of role and conversation features from the first task, we also re-implement the conversational-statistics feature set described in the original work (details in Appendix D.1) and combine it with role features. Adding roles consistently improves performance over conversational statistics alone, and the best result comes from team-context roles + conversational statistics (0.74), exceeding the best reported combination of prior feature families (0.70). We also include a zero-shot LLM baseline that predicts performance gain directly from the group dialogue tran-

script (detailed prompts in Appendix D.2), which is competitive (0.68) but does not match the best role + conversation features model.

7 Discussion

Our results position theory-grounded communication roles as interpretable, mid-level representations of teamwork. In this section, we discuss implications for role-based support in team-based education and, more broadly, for socially aware language technologies.

Role-based Chat and Learning Assistants Detecting roles from team chat enables role-aware feedback that encourages team members to enact missing roles, which is especially important in collaborative learning (He et al., 2023). For example, when a team contains many *explorers* who question but few *facilitators* who control the discussion, a role-aware chat assistant could prompt the team to summarize decisions or propose concrete next steps. Similarly, when participation appears uneven, it could encourage *gatekeeper*-style check-ins (e.g., inviting input from quieter members). The same signals may also be used to assess and teach teamwork skills based on a student's long-term communication history.

Designing Socially Aware Agents Instead of relying on fixed personas or latent behaviors, agents can be guided to adopt situational communicative roles (e.g., shifting into facilitation when coordination breaks down) while preserving human ownership of the collaboration. In single human-agent interactions, team roles can help create specific types of agents for a particular task, similar to shaping their personality (Li et al., 2025b) but specifically as "teammates"; whereas in agent-agent interactions (e.g., ChatDev, Qian et al., 2024), our findings that humans tend to take on multiple roles also motivate encouraging role diversity rather than homogeneous optimization (e.g., pairing initiating and facilitating behaviors). Future work can extend our study to both human-agent and agent-agent interactions, empirically testing whether assigning specific role combinations leads to higher performance in real-life or synthetic teams.

Multimodal Role Modeling Chat logs capture only one facet of collaboration. While communication is an important dimension of collaboration, it misses other crucial parts of teamwork. For example, in an engineering classroom, a student labeled as an *outsider* in chat might be a "silent workhorse" contributing heavily to the codebase. Future work can combine chat history with signals in other collaboration channels and modalities when modeling team roles, such as log data from digital collaborative systems (e.g., GitHub commits and code reviews, Song et al., 2026). Combining communication with action-based evidence could enable more accurate role inference that aligns communicative functions with technical actions.

8 Conclusion

In this work, we operationalize a theory-grounded taxonomy of eight communication roles to study the dynamics of teamwork. By leveraging LLMs for scalable annotation, we characterize how team members enact and shift roles across the lifecycle of a semester-long project. We demonstrate that these role representations capture critical collaborative signals, outperforming standard lexical, conversational, and zero-shot LLM baselines in predicting both peer recognition and team performance. Ultimately, our findings highlight the value of tracking communication roles to better understand, evaluate, and support effective collaboration. Future work can combine message data with additional behavioral traces (e.g., code/document edit logs) to better capture off-platform contributions and design role-aware interventions or agents for human and socially aware AI teammates.

Limitations

While we validate the role constructs on a second external benchmark (DeliData), our primary dataset comes from a single computer science course at a North American university. The relationships between roles and peer recognition in our course setting may differ across domains, cultures, and communication channels. Our dataset also includes only students who consented and teams that actively used the course-provisioned Slack channel, which may introduce selection effects, underrepresenting negative or sensitive interactions.

In addition, although LLM annotation achieves high agreement with experts in our setting, results depend on prompting choices and proprietary reasoning models for best performance, and model biases related to language style may propagate into role labels (e.g., polite messages are more likely to be labeled for constructive roles). More broadly, role detection could be misused for surveillance or

high-stakes evaluation. Specifically in the educational context, we intend to use it as a formative lens for team or individual reflection, rather than an automated grading tool.

Acknowledgments

This work was supported in part by the Strategic Instructional Innovations Program (SIIP) in the Grainger College of Engineering at the University of Illinois Urbana-Champaign (UIUC). We thank our colleagues in the Language Interaction Lab and the ORCHID Lab at UIUC for their invaluable feedback. We also thank the students who shared their data for this study and the anonymous reviewers for their helpful comments.

References

Kelsey Allen, Giuseppe Carenini, and Raymond Ng. 2014. Detecting disagreement in conversations using pseudo-monologic rhetorical structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1169–1180, Doha, Qatar. Association for Computational Linguistics.

A. Aranzabal, E. Epelde, and M. Artetxe. 2022. Team formation on the basis of belbin's roles to enhance students' performance in project based learning. *Education for Chemical Engineers*, 38:22–37.

Tal August, Lauren Kim, Katharina Reinecke, and Noah A. Smith. 2020. Writing strategies for science communication: Data and computational analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5327–5344, Online. Association for Computational Linguistics.

Satanjeev Banerjee and Alexander Rudnicky. 2006. You are what you say: Using meeting participants' speech to detect their roles and expertise. In *Proceedings of the Analyzing Conversations in Text and Speech*, pages 23–30, New York City, New York. Association for Computational Linguistics.

Kenneth D. Benne and Paul Sheats. 1948. Functional roles of group members. *Journal of Social Issues*, 4(2):41–49. Place: United Kingdom Publisher: Blackwell Publishing.

Hancheng Cao, Vivian Yang, Victor Chen, Yu Jin Lee, Lydia Stone, N'godjigui Junior Diarrassouba, Mark E. Whiting, and Michael S. Bernstein. 2021. My team will go on: Differentiating high and low viability teams through team interaction. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW3).

Wen Dong, Bruno Lepri, Fabio Pianesi, and Alex Pentland. 2013. Modeling functional roles dynamics in small group interactions. *IEEE Transactions on Multimedia*, 15(1):83–95.

Raji Ghawi, Siegfried Müller, and Jürgen Pfeffer. 2021. Improving team performance prediction in mmogs with temporal communication networks. *Social Network Analysis and Mining*, 11.

Shanyun He, Xinyue Shi, Tae-Hee Choi, and Junqing Zhai. 2023. How do students'roles in collaborative learning affect collaborative problem-solving competency? a systematic review of research. *Thinking Skills and Creativity*, 50:101423.

Christopher Hundhausen, Phill Conrad, Olusola Adesope, Ahsun Tariq, Samir Sbai, and Andrew Lu. 2023. Investigating reflection in undergraduate software development teams: An analysis of online chat transcripts. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, SIGCSE 2023, page 743–749, New York, NY, USA. Association for Computing Machinery.

Farnaz Jahanbakhsh, Wai-Tat Fu, Karrie Karahalios, Darko Marinov, and Brian Bailey. 2017. You want me to work with who? stakeholder perceptions of automated team formation in project-based courses. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 3201–3212, New York, NY, USA. Association for Computing Machinery.

Chenyan Jia, Michelle S. Lam, Minh Chau Mai, Jeffrey T. Hancock, and Michael S. Bernstein. 2024. Embedding democratic values into social media ais via societal objective functions. *Proc. ACM Hum.-Comput. Interact.*, 8(CSCW1).

Georgi Karadzhov, Tom Stafford, and Andreas Vlachos. 2023. Delidata: A dataset for deliberation in multiparty problem solving. *Proceedings of the ACM on Human-Computer Interaction*, 7(CSCW2):1–25.

Klaus Krippendorff. 2004. Reliability in content analysis: Some common misconceptions and recommendations. *Human Communication Research*, 30(3):411–433.

Yihuai Lan, Zhiqiang Hu, Lei Wang, Yang Wang, Deheng Ye, Peilin Zhao, Ee-Peng Lim, Hui Xiong, and Hao Wang. 2024. LLM-based agent society investigation: Collaboration and confrontation in avalon gameplay. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 128–145, Miami, Florida, USA. Association for Computational Linguistics.

Haoran Li, Ziyi Su, Yun Xue, Zhiliang Tian, Yiping Song, and Minlie Huang. 2025a. Advancing collaborative debates with role differentiation through multi-agent reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22655–22666, Vienna, Austria. Association for Computational Linguistics.

Wenkai Li, Jiarui Liu, Andy Liu, Xuhui Zhou, Mona T. Diab, and Maarten Sap. 2025b. BIG5-CHAT: Shaping LLM personalities through training on human-grounded data. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20434–20471, Vienna, Austria. Association for Computational Linguistics.

Diane Litman, Susannah Paletz, Zahra Rahimi, Stefani Allegretti, and Caitlin Rice. 2016. The teams corpus and entrainment in multi-party spoken dialogues. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1421–1431, Austin, Texas. Association for Computational Linguistics.

Keming Lu, Bowen Yu, Chang Zhou, and Jingren Zhou. 2024. Large language models are superpositions of all characters: Attaining arbitrary role-play via self-alignment. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7828–7840, Bangkok, Thailand. Association for Computational Linguistics.

Keith Maki, Michael Yoder, Yohan Jo, and Carolyn Rosé. 2017. Roles and success in Wikipedia talk pages: Identifying latent patterns of behavior. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1026–1035, Taipei, Taiwan. Asian Federation of Natural Language Processing.

John Mathieu, John Hollenbeck, Daan Knippenberg, and Daniel Ilgen. 2017. A century of work teams in the journal of applied psychology. *Journal of Applied Psychology*, 102:452–467.

John E. Mathieu, Scott I. Tannenbaum, Michael R. Kukenberger, Jamie S. Donsbach, and George M. Alliger. 2015. Team role experience and orientation: A measure and tests of construct validity. *Group & Organization Management*, 40(1):6–34.

R. Meredith Belbin. 2011. Management Teams: Why They Succeed or Fail (3rd ed.). *Human Resource Management International Digest*, 19(3).

Kristina Nestsiarovich and Dirk J. Pons. 2020. Team role adoption and distribution in engineering project meetings. *Behavioral Sciences*, 10.

Thien Hai Nguyen, Kiyoaki Shirai, and Julien Velcin. 2016. Sentiment analysis on social media for stock movement prediction. *Expert Syst. Appl.*, 42(24):9603–9611.

Vlad Niculae and Cristian Danescu-Niculescu-Mizil. 2016. Conversational markers of constructive discussions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–578, San Diego, California. Association for Computational Linguistics.

Mazni Omar, Sharifah-Lailee Syed-Abdullah, and Naimah Mohd Hussin. 2011. Developing a team performance prediction model: A rough sets approach. In *Informatics Engineering and Information Science*, pages 691–705, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand. Association for Computational Linguistics.

David Reitter and Johanna D. Moore. 2007. Predicting Success in Dialogue. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 808–815, Prague, Czech Republic. Association for Computational Linguistics.

Antonette Shibani, Elizabeth Koh, Vivian Lai, and Kyong Jin Shim. 2017. Assessing the language of chat for teamwork dialogue. *Journal of Educational Technology Society*, 20(2):224–237.

Yifan Song, Ritika Vithani, Wenxuan Wendy Shi, and Brian P. Bailey. 2026. From data to action: Empowering students to assess and improve teamwork with cross-tool log data. In *Proceedings of the 57th ACM Technical Symposium on Computer Science Education V.1*, page 992–998, New York, NY, USA. Association for Computing Machinery.

P. C. Wason. 1968. Reasoning about a rule. *Quarterly Journal of Experimental Psychology*, 20(3):273–281.

Anna Wegmann, Tijs A. Van Den Broek, and Dong Nguyen. 2024. What's mine becomes yours: Defining, annotating and detecting context-dependent paraphrases in news interview dialogs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 882–912, Miami, Florida, USA. Association for Computational Linguistics.

Diyi Yang, Miaomiao Wen, and Carolyn Rosé. 2015. Weakly supervised role identification in teamwork interactions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1671–1680, Beijing, China. Association for Computational Linguistics.

Justine Zhang, Jonathan Chang, Cristian Danescu-Niculescu-Mizil, Lucas Dixon, Yiqing Hua, Dario Taraborelli, and Nithum Thain. 2018. Conversations Gone Awry: Detecting Early Signs of Conversational Failure. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1350–1361, Melbourne, Australia. Association for Computational Linguistics.

A Full Adapted Role Taxonomy

We adapt the education-grounded team-role taxonomy of Nestsiarovich and Pons (2020) to Slack-based project coordination by clarifying boundary cases (e.g., routine scheduling does not count toward work roles; internal technical links do not count as *connector*). Table 8 lists the full guidelines used for expert annotation.

B LLM Role Annotation Details

B.1 Role annotation prompt

We use zero-shot prompting to label the presence of each of the eight constructive roles for a student–deliverable instance (Section 4).

System prompt template. The system message instructs the model to make role-by-role binary decisions using only explicit textual evidence in the provided messages, and to cite evidence by message indices. The following prompt skips the communication patterns of the roles, which are identical to Appendix Table 8.

```
You are an expert annotator labeling team roles using only the student's Slack messages in one deliverable period of a semester-long team project.
Work role-by-role. For each role, choose the final label (0 or 1) for whether the student demonstrated that role in this deliverable period and provide brief reasons (1–2 sentences) for positive labels referencing specific message numbers (e.g., [3], [10]) as evidence.
Do not guess from tone, use clear textual actions (e.g., asking, proposing, summarizing, mediating, representing, inviting, connecting). Multi-label allowed. Label ambiguous ones as 0.
Roles:
« ROLE_DEFS, see Appendix Table 8 »
```

User prompt template. The user prompt includes all messages authored by one student within one deliverable window, indexed in chronological order. To reduce formatting errors, we request a structured JSON output with a fixed schema.

```
Student messages (indexed and anonymized):
{[1] xxx [2] xxx ... [N] xxx}
Return strict JSON matching the schema.
```

Output format. The JSON object contains exactly eight keys (one per role). Each key maps to an object with an integer `label` $\in \{0, 1\}$ and a short `reason`.

```
{
"Initiator": {"label": 0/1, "reason": "..."},
"Explorer": {"label": 0/1, "reason": "..."},
. . .
"Connector": {"label": 0/1, "reason": "..."}
}
```

B.2 Inference settings and API parameters

We run role annotation via the OpenAI API for GPT models and via OpenRouter API for Gemini and DeepSeek models. Table 9 summarizes the inference settings used for LLM role annotation. Unless noted below, we use API default values for the parameters.

B.3 Macro performance across LLM variants

We compare LLM-generated role labels against expert annotations on the same expert-labeled set. Table 10 reports macro-averaged F1 and Krippendorff's α. Across families, larger and reasoning-capable variants tend to perform better, though increasing reasoning effort is not always beneficial.

C Peer Recognition Experiment Details

C.1 Supervised classifier configuration

All supervised peer recognition models use `scikit-learn` Logistic Regression with default L2 regularization.

For structured feature vectors (conversational, roles, roles+conversational), we standardize features and fit logistic regression:

```
make_pipeline(
StandardScaler(with_mean=True),
LogisticRegression(solver="lbfgs", max_iter=3000,
class_weight="balanced", random_state=42)
)
```

For BoW, we use `CountVectorizer` with `ngram_range=(1,3)` followed by logistic regression:

```
make_pipeline(
CountVectorizer(ngram_range=(1,3)),
LogisticRegression(max_iter=2000,
class_weight="balanced")
)
```

C.2 Zero-shot LLM baseline prompts

We evaluate a zero-shot LLM baseline that predicts whether an instance satisfies a given AF threshold. We run one independent prompt per split: penalty, bonus, or average.

Role	Typical communication pattern
Initiator	Active participation, proposes new ideas and tasks, and introduces new directions of work. Scheduling or organizing meetings do not count.
Explorer	Active data collecting: asks general questions; requests facts, ideas, or opinions; explores alternatives; asks to clarify or specify ideas, define terms, and provide examples. Routine scheduling questions do not count.
Information Provider	Provides detailed and extensive information: takes an active part in the conversation, but mostly talks rather than listens. Providing scheduling availability does not count.
Facilitator	Defines the task or problem; suggests a method or process for accomplishing the task; provides structure; controls discussion processes; brings the group back on track. Organizing meetings may count but only when the student actively drives coordination, not merely availability updates.
Arbitrator	Encourages the group to find agreement whenever a miscommunication arises or when the group cannot come to a common position.
Representative	Verbalizes the group's feelings, hidden problems, questions, or ideas that others are afraid to express; answers questions referred to the whole group.
Gatekeeper	Helps keep communication channels open: fills gaps in conversation; asks a person for their opinion; is sensitive to signals indicating that people want to participate.
Connector	Connects the team with people outside the group. Internal links or purely technical resources such as docs/libraries do not count.
Passive Collector	Passive data collecting: non-verbal signs of agreement or short yes/no answers; low verbal participation in team discussion; attentive listening; keeping ideas inside (non-vocalization).
Outsider	Does not participate in project discussion.

Table 8: Full adapted team role taxonomy and role descriptions from Nestsiarovich and Pons (2020). We retain the eight constructive roles (top block) and exclude *passive collector* and *outsider* for modeling in the main paper. The communication patterns are used as annotation guidelines for both human labeling and LLM prompting.

Setting	Value
Task	Role annotation (8-way multi-label; binary per role)
API	OpenAI API for GPT series and OpenRouter API for Gemini and DeepSeek series
Input unit	One participant's messages within a deliverable window
Output format	JSON Schema with fields `label` and `reason` per role
Max output tokens	`max_output_tokens = 5000`
Reasoning Setting (detailed model variants in Appendix Table 10)	
For GPT	`Effort` can be set to `none, minimal, low, medium, high`
For Gemini	`Effort` can be set to `minimal, low, medium, high`
For DeepSeek	Can choose either `thinking.enabled` or `thinking.disabled`

Table 9: LLM role-annotation inference settings; unmentioned parameters used default values

System prompt template. `You are an expert in team communication analytics. Given one student's Slack message history, predict the probability of the following outcome for their within-team peer evaluation score: {CONDITION: definition of penalty, bonus, or above average, see Section 5.1}.`
`Respond strictly as JSON with keys "prob" (0-1).`

User prompt template. `Messages:`
`{ANONYMIZED_RAW_TEXT}`

D DeliData Experiment Details

D.1 Conversational statistics features

We re-implement the conversational-statistics feature set following Appendix A.3 of the original paper (Karadzhov et al., 2023) since the authors didn't publish their code.

For each group dialogue transcript (9 features):

- number of participants in the chat
- total number of messages
- average number of messages per player
- average number of tokens per player
- total unique tokens
- average unique tokens per player
- participants' individual performance
- diversity in participants' individual solutions
- group consensus

D.2 Zero-shot LLM baseline prompt for performance gain

System prompt. `You are a team performance analyst. Given a team discussion transcript from`

Model	Macro F1	Macro Krippendorff's α
GPT-5-mini (`reasoning={"effort":"minimal"}`)	0.614	0.422
GPT-5.1 (`reasoning={"effort":"none"}`)	0.706	0.600
GPT-5.1 (`reasoning={"effort":"low"}`)	**0.787**	**0.723**
GPT-5.1 (`reasoning={"effort":"high"}`)	0.761	0.671
Gemini 3 Flash (`reasoning={"effort":"minimal"}`)	0.688	0.496
Gemini 3 Pro (`reasoning={"effort":"low"}`)	0.727	0.576
Gemini 3 Pro (`reasoning={"effort":"high"}`)	0.743	0.634
DeepSeek V3.1 (`thinking={"type":"disabled"}`)	0.600	0.455
DeepSeek V3.1 (`thinking={"type":"enabled"}`)	0.647	0.526

Table 10: Macro performance across LLM variants for role annotation on the expert-labeled set (N = 424). GPT-5.1 (`reasoning={"effort":"low"}`) yields the best performance.

```
the Wason card task,
predict whether the team's final performance
improved compared to their initial performance.
Output a JSON object with a single key "prob_gain"
between 0 and 1.
```

User prompt. `Transcript:`

```
{TEAM_TRANSCRIPT}
```