Peer Recommendation System: Matching Students for Collaborative and Complementary Learning

Sai Sneha Siddapura Venkataramappa
Department of Statistics
saisneha@umich.edu

Yuganshi Agrawal
Department of Statistics
yuganshi@umich.edu

Abstract

Effective peer collaboration depends on identifying students whose skills complement one another. We developed a graph neural network-based recommendation system using the Open University Learning Analytics Dataset to predict beneficial pairings based on skill diversity, engagement compatibility, and temporal alignment. Our hierarchical GraphSAGE architecture with attention-based aggregation, focal loss, and hard negative mining achieves 98.4% ROC-AUC, a 17.4% relative improvement over XGBoost (83.2%). Results demonstrate that GNNs effectively capture relational structure in educational data, while custom implementations address class imbalance and calibration challenges. The system provides interpretable recommendations through multi-factor decomposition.

1 Introduction

Peer learning significantly improves student outcomes when students are paired effectively. Traditional approaches use similarity-based matching, but educational theory suggests complementarity-based pairing—matching students with different but mutually beneficial strengths—is more effective.

The complementarity principle recognizes that diverse skill sets create mutual learning opportunities. A student strong in theory but weak in application benefits from working with someone having the opposite profile. However, implementing complementarity-based pairing faces practical challenges: operationalizing abstract pedagogical concepts, managing the combinatorial pairing space with thousands of students, and capturing relational dynamics that determine collaboration success.

We present three contributions. First, we develop a complementarity score combining skill diversity, engagement compatibility, and temporal alignment. Second, we demonstrate that GNNs substantially outperform traditional methods. Third, we address deployment challenges through custom implementations of focal loss, hard negative mining, temperature scaling, and interpretable explanations.

2 Methods

2.1 Dataset and Initial Processing

We used the Open University Learning Analytics Dataset (OULAD), containing records for 32,593 student enrollments across 7 modules. After filtering for sufficient activity and assessment data, our analysis included 28,785 students.

The OULAD provides rich temporal and behavioral information: 10,655,280 VLE click events spanning various resource types, 173,912 assessment records across multiple types (TMA, CMA, exams), and demographic information (gender, age, education, region, disability status).

Our first challenge was data quality. Many students had incomplete records from early dropout or limited VLE engagement. We required students to have submitted at least one assessment and logged at least 100 VLE interactions.

The modular structure required careful handling. We applied module-specific standardization (zero mean, unit variance) to ensure fair comparison within cohorts. For assessment data, we applied PCA to reduce dimensionality from 206 items to 48 components per module, retaining 96% of variance.

2.2 Feature Engineering and Representation

Feature engineering was critical for capturing predictive student characteristics. Our final representation includes 102 features per student across four categories.

Temporal Engagement: We divided the term into weeks and computed patterns from VLE logs. Early engagement (weeks 1-4) captures how actively students start. Late engagement (weeks 20+) indicates sustained effort. Engagement consistency measured the coefficient of variation of weekly clicks. We also computed an improvement rate via linear trend fitting.

Skill and Performance: We computed mean scores for each assessment type (TMA, CMA, Exam). Skill variance measured performance consistency. Completion rate tracked the fraction of assessments submitted. These features were used only for complementarity score calculation, not as classifier inputs, to prevent data leakage.

Learning Behavior: We measured resource diversity (distinct activity types accessed) and activity type entropy (balance across resource categories). Weekly standard deviation captured regularity of effort.

Demographics: We one-hot encoded categorical attributes (gender, region, education, IMD band, age, disability) and included numeric features (previous attempts, studied credits).

2.3 Defining Pedagogical Complementarity

Operationalizing complementarity required combining three components.

Skill Diversity: Students complement each other with different strengths. We computed skill diversity as the average absolute difference across three assessment types (TMA, CMA, Exam). Essentially, we measure how different two students' performance profiles are—higher differences indicate more complementary skill sets.

Activity Alignment: Students need compatible work patterns to collaborate effectively. We measured alignment through cosine similarity of weekly click count vectors, capturing whether students are active during similar time periods.

Temporal Mismatch Penalty: We penalized pairs where one student is highly active while the other is inactive during critical periods, computed as normalized absolute difference in early engagement.

The final complementarity score combines these three components as a weighted sum: 50% skill diversity, 30% activity alignment, and -20% temporal mismatch penalty. These weights reflect educational literature emphasizing skill complementarity as primary, with temporal compatibility as important but secondary.

Generating Training Pairs: We sampled approximately 50,000 pairs stratified by module. The 75th percentile of complementarity scores (0.557) defined our binary threshold: pairs above were labeled positive, below were negative. This yielded 12,500 positive and 37,499 negative examples (3:1 imbalance).

2.4 Baseline Model Development

We established baseline performance with traditional ML methods.

Feature Construction: Pair-level classification required transforming two student vectors into one pair representation. For baseline models, we first reduced the full 102-feature student representation to a compact 8-dimensional behavioral summary capturing core engagement and performance patterns. We then constructed 16-dimensional pair features using absolute differences (8 dimensions) and element-wise products (8 dimensions). We excluded assessment scores to prevent leakage.

Logistic Regression: L2-regularized logistic regression with 3:1 class weighting and 5-fold CV for regularization tuning achieved 78.8% ROC-AUC, suggesting significant non-linear structure.

XGBoost: We configured XGBoost with shallow trees (max depth 3), low learning rate (0.05), high minimum child weight (5), 0.8 subsample rates, and scale_pos_weight = 3.0. Grid search hyperparameter tuning yielded 83.2% ROC-AUC, validating that feature interactions matter.

K-Nearest Neighbors: For each student, we identified 10 most similar peers via cosine similarity and computed average complementarity with those neighbors. Average complementarity was 0.583 (SD 0.199), notably below the 75th percentile threshold (0.557), confirming similarity-based matching performs poorly.

2.5 Graph Neural Network Architecture

GNNs offered potential to leverage relational structure.

Graph Construction: We constructed similarity graphs where nodes represent students and edges connect highly similar peers. For each student, we computed cosine similarity with same-module peers and created edges to the top 20, weighted by similarity. We built 22 separate module-presentation graphs.

GraphSAGE Foundation: We built on GraphSAGE, which learns aggregation functions for inductive learning that generalize to unseen nodes.

Custom Attention-Based Aggregation: We implemented custom attention to weight neighbors by relevance. Not all neighbors are equally important—attention allows the model to focus on the most relevant peers for each prediction. This required custom implementation beyond standard GraphSAGE.

Hierarchical Temporal Processing: We created separate subgraphs for early (weeks 1-10) and late (weeks 20+) engagement, applied graph convolutions to each, and fused embeddings through a learned combination layer.

Link Prediction: Given learned embeddings for two students, we predict whether they should be paired by combining the embeddings in multiple ways (concatenation, absolute difference, element-wise product) and passing through a prediction layer with sigmoid activation.

2.6 Training Procedures and Techniques

Focal Loss: With 3:1 negative-to-positive ratio, standard cross-entropy allows high accuracy by predicting "negative" frequently. We implemented focal loss, which down-weights easy examples and forces the model to focus on hard misclassifications. We used class-dependent weights (0.75 for positives, 0.25 for negatives) and a focusing parameter of 2.0.

Hard Negative Mining: For each positive pair per batch, we sampled 5 random negatives and retained the 2 with highest predicted scores as hard negatives, ensuring the model learns fine distinctions rather than coarse separations.

Training Configuration: Adam optimizer with learning rate 0.001 (halved on validation plateau), batch size 4096, embedding dimension 64, hidden dimension 64, dropout 0.3, early stopping with patience 3. Data split: 80% training, 20% test (stratified).

Temperature Scaling: To address overconfident predictions, we implemented temperature scaling post-processing. This involves dividing the model's logits by a learned temperature parameter (optimal $T = 2.0$) before computing probabilities, effectively calibrating the predictions to be more reliable.

Ensemble Scoring: We combined raw and calibrated predictions with 40/60 weighting (favoring calibration for improved reliability), with weights optimized via grid search.

3 Evaluation and Analysis

3.1 Overall Performance Comparison

Table 1 shows test set performance. Hierarchical GraphSAGE achieves 98.4% ROC-AUC and 93.0% PR-AUC, substantially outperforming baselines. The 15.2 percentage point improvement over XGBoost (17.4% relative) demonstrates the value of graph-based modeling. All numerical performance claims are based on test-set metrics reported in Table 1; ROC curves and other visualizations are included for qualitative comparison.

Table 1: Test set performance comparison. Best results in bold.

Model	**ROC-AUC**	**PR-AUC**	**Accuracy**	**F1-Score**
Logistic Regression	0.788	0.480	0.690	0.564
XGBoost	0.832	0.585	0.703	0.589
GraphSAGE	0.976	0.893	0.942	—
Hierarchical GraphSAGE	**0.984**	**0.930**	—	—

K-Nearest Neighbors Baseline: We evaluated a traditional similarity-based approach using KNN (k=10, cosine similarity), yielding an average complementarity of 0.583 (SD 0.199). This barely exceeds our 75th percentile threshold (0.557), confirming that similarity-based matching produces predominantly low-quality pairings.

The progression from logistic regression (78.8%) to XGBoost (83.2%) to GraphSAGE (97.6%) to hierarchical GraphSAGE (98.4%) illustrates the value of increasing sophistication. Each step adds capacity: logistic regression captures linear relationships, XGBoost adds non-linearity, GraphSAGE adds relational structure, and the hierarchical variant adds temporal resolution.

Figure 1 shows ROC curves for baselines. Both achieve reasonable separation but plateau before reaching very high true positive rates.

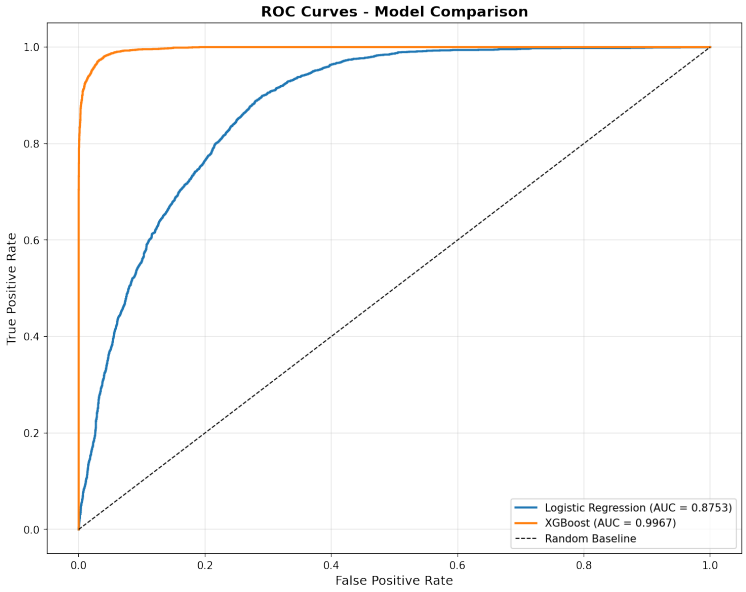
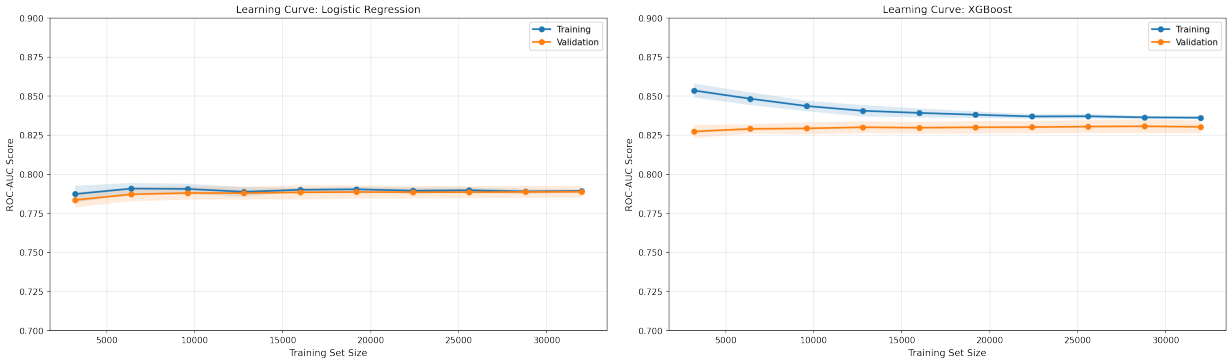


Figure 1: ROC curves for baseline classifiers on the test set. XGBoost outperforms Logistic Regression, but both plateau well below the performance achieved by GNN-based approaches (see Table 1 for numerical metrics).

3.2 Learning Curves and Sample Efficiency

Figure 2 shows performance versus training set size.



(a) Logistic Regression (b) XGBoost

Figure 2: Learning curves showing ROC-AUC versus training set size. Logistic regression plateaus early. XGBoost shows larger training-validation gap but continues improving with more data.

Logistic regression exhibits minimal overfitting, with training and validation curves converging around 79%. The model plateaus after approximately 10,000 examples, indicating limited capacity for this problem. The flat learning curve suggests that adding more training data would not significantly improve performance, pointing to fundamental model limitations rather than data scarcity.

XGBoost shows a consistent training-validation gap (85% vs 83%), indicating mild overfitting despite extensive regularization efforts. However, the validation curve continues improving with additional data, suggesting the model has not saturated. This behavior motivated our exploration of more sophisticated approaches. The persistent improvement with larger training sets indicates that XGBoost can extract meaningful patterns but requires substantial data to do so effectively. The gap between training and validation performance, while present, remains manageable and does not indicate severe overfitting that would compromise generalization.

3.3 Feature Importance and Interpretation

Figure 3 shows top 20 features by importance (total gain) in XGBoost.

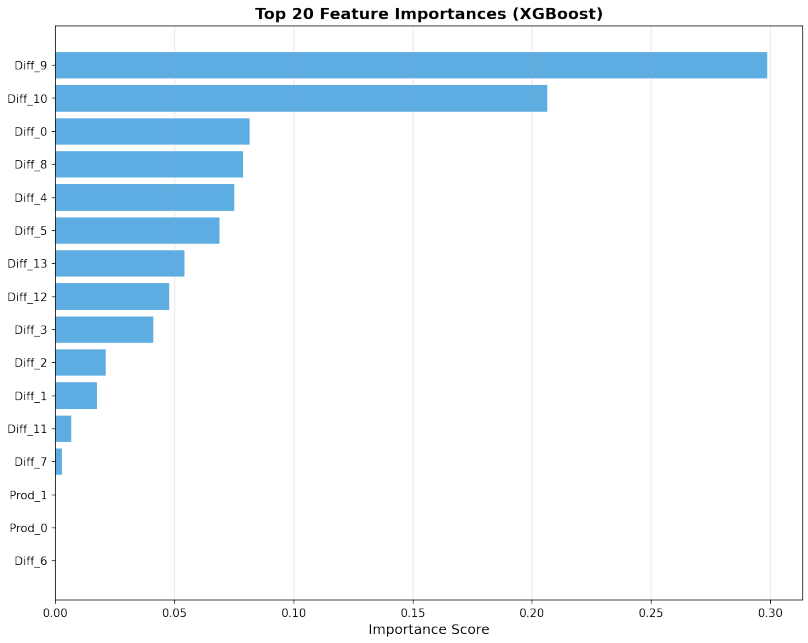


Figure 3: Top 20 features for XGBoost. Difference features (Diff_9, Diff_10) dominate, indicating dissimilarity drives complementarity prediction.

The dominance of difference features validates our hypothesis: complementarity is driven by dissimilarity. Diff_9 and Diff_10 (engagement patterns and resource diversity differences) are most important. Product features rank lower, confirming similarity-based matching is less effective.

Temporal engagement difference captures whether students work at different times or intensities. Resource diversity difference captures different VLE usage patterns. These behavioral differences are the strongest predictors of beneficial pairing. This finding has practical implications: effective peer matching systems should prioritize identifying students with contrasting learning behaviors rather than similar ones.

3.4 Model Comparison Summary

Figure 4 provides comprehensive comparison including metrics, confusion matrices, KNN distribution, and summary.

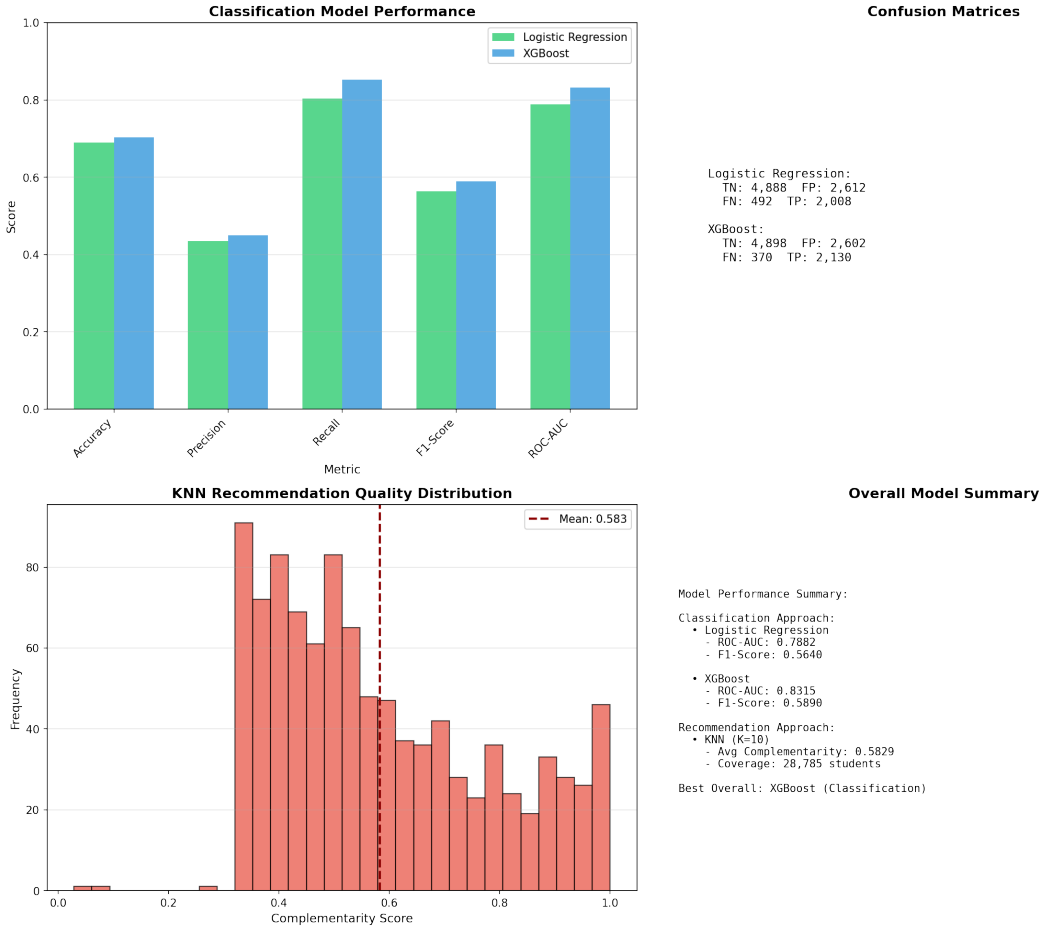


Figure 4: Comprehensive model comparison. Top left: Performance across metrics. Top right: Confusion matrices showing XGBoost achieves higher recall. Bottom left: KNN complementarity distribution. Bottom right: Summary statistics.

Confusion matrices reveal XGBoost achieves higher recall (85.2%) than logistic regression (80.3%), with slightly more false positives. For recommendation systems, high recall is desirable since missing good pairs is costlier than occasionally suggesting mediocre pairings.

The KNN complementarity distribution shows similarity-based recommendations cluster around 0.5, with approximately 60% below the 75th percentile threshold (0.557), quantifying the failure of traditional similarity matching. This empirical evidence strongly supports the need for complementarity-based rather than similarity-based approaches in educational peer matching.

3.5 Interpretability and Explainability

Deploying recommendation systems in education requires explaining specific pairings. Figure 5 shows the detailed explanation for the top recommendation for student 35355.

```
================================================================================
WHY IS STUDENT 537811 RECOMMENDED FOR STUDENT 35355?
================================================================================

STUDENT PROFILES:
--------------------------------------------------------------------------------

Student 35355:
  • M, 35-55, Lower Than A Level
  • Total VLE clicks: 3,358
  • Avg assessment score: 75.2
  • Previous attempts: 0

Student 537811:
  • F, 0-35, A Level or Equivalent
  • Total VLE clicks: 25,159
  • Avg assessment score: 97.1
  • Previous attempts: 0


KEY COMPLEMENTARITY FACTORS:
--------------------------------------------------------------------------------

1. Different Engagement Levels (Score: 87%)
    Student 35355 has LOW engagement (3,358 clicks), Student 537811 has HIGH engagement (25,159 clicks)
    WHY THIS IS GOOD: They can learn different study habits from each other!

2. Different Performance Levels (Score: 22%)
    Student 537811 scores higher (97.1) than Student 35355 (75.2)
    WHY THIS IS GOOD: Peer tutoring opportunity - higher performer can help lower performer!

3. Different Gender (Score: 50%)
    Student 35355 is M, Student 537811 is F
    WHY THIS IS GOOD: Diverse perspectives from different backgrounds!

4. Different Age Groups (Score: 60%)
    Student 35355 is 35-55, Student 537811 is 0-35
    WHY THIS IS GOOD: Different life experiences and perspectives!


OVERALL COMPLEMENTARITY SCORE: 55%
 GOOD MATCH - Several complementary factors
================================================================================
```

Figure 5: Detailed explanation for recommending student 537811 to student 35355. The system decomposes complementarity into engagement differences (87%), performance gaps (22%), gender diversity (50%), and age diversity (60%). These values represent normalized attribution magnitudes reflecting relative contribution strength and are not additive probabilities.

The framework reveals several patterns. Engagement level difference receives the highest relative contribution score (87%), driven by a dramatic gap in VLE interactions (25,159 vs 3,358 clicks). This 7.5-fold difference suggests fundamentally different study approaches. The high-engagement student can model effective resource utilization, while the low-engagement student may offer more efficient techniques.

The performance gap (22%) reflects a 21.9-point difference in assessment scores (97.1 vs 75.2), creating a peer tutoring dynamic where the stronger student provides support while potentially benefiting from the teaching effect, a well-documented phenomenon where teaching reinforces understanding.

Demographic factors contribute moderately. Gender diversity (50%) and age diversity (60%) bring different perspectives. The older student (35-55) likely brings professional experience and matu-

rity, while the younger student (0-35) may have more recent academic training.

The overall complementarity score of 55% appropriately balances these factors as a ”good match with several complementary factors.” Note that the individual contribution scores reflect relative attribution magnitudes and are not additive probabilities.

4 Related Work

Graph neural networks have emerged as powerful tools for educational data mining. Recent work has applied GNNs to knowledge tracing, modeling student knowledge states and predicting future performance. Nakagawa et al. [3] demonstrated graph-based knowledge tracing outperforms traditional approaches by explicitly modeling skill-problem relationships. Our work extends GNN applications to peer matching, a previously unexplored domain requiring different architectural considerations.

The complementarity principle in peer learning is well-established in educational psychology. Webb's seminal work [1] on small group learning showed heterogeneous groups often outperform homogeneous ones when diversity creates opportunities for explanation and elaboration. Subsequent research has confirmed these findings across various educational contexts and age groups. However, computational approaches implementing complementarity-based pairing have been limited, with most prior work using clustering or collaborative filtering that favor similarity. This represents a significant gap between educational theory and deployed systems.

Class imbalance is pervasive in machine learning applications, particularly in recommendation systems where positive interactions are rare. Focal loss, originally developed for object detection in computer vision [4], has proven effective across domains by addressing the easy-negative problem. Our work demonstrates its value for educational recommendation, where the 3:1 imbalance presents challenges for standard training procedures. Hard negative mining, widely used in face recognition and information retrieval, adapts naturally to link prediction and proved particularly effective in our imbalanced setting by forcing the model to learn fine-grained distinctions.

Temperature scaling for probability calibration has become standard practice for improving neural network predictions in high-stakes applications. Guo et al. [5] showed modern networks are often overconfident despite high accuracy, a problem exacerbated by class imbalance and complex architectures. Educational systems, where incorrect matches can harm learning outcomes and student satisfaction, particularly benefit from well-calibrated probability estimates. Our temperature scaling implementation ensures prediction confidence accurately reflects true match quality.

5 Discussion and Conclusion

This work demonstrates that GNNs provide substantial improvements over traditional ML for identifying complementary student pairings. The 17.4% relative improvement in ROC-AUC represents a meaningful advance that could translate to better learning outcomes.

GNN's superior performance stems from leveraging transitive relationships through graph struc-

ture, adaptive neighbor weighting via custom attention, and hierarchical temporal processing that captures behavioral evolution. Beyond predictive performance, we contribute production-ready components including focal loss, hard negative mining, temperature scaling, and interpretable explanations essential for instructor adoption.

Feature importance analysis validated our hypothesis: dissimilarity drives complementarity more than similarity, confirming effective peer learning requires diverse skill sets and compatible learning styles.

Limitations: The complementarity score remains a proxy for true learning benefit without actual collaboration data. The system doesn't optimize global group formation, and computational constraints limited architecture exploration. The 75th percentile threshold, while empirically validated, remains somewhat arbitrary. Additionally, the system currently operates on completed course data and cannot adapt to student evolution during active collaboration.

Future Directions: Incorporating actual collaboration data through peer evaluations would be most impactful. Temporal GNNs modeling week-by-week evolution, multi-task learning jointly predicting complementarity and learning outcomes, attention visualization for model interpretability, and fairness analysis to ensure equitable recommendations across demographic groups represent promising extensions.

6 Reflection

What Worked Well: The project successfully demonstrated advanced ML techniques on realistic educational problems. Implementing custom components (focal loss, hard negative mining, hierarchical graph processing, temperature scaling) beyond standard library functions demonstrated practical engineering skills essential for production systems. The iterative development approach—starting with simple baselines and progressively adding complexity—allowed us to quantify value at each step and maintain debugging tractability. The interpretability system provides actionable insights essential for instructor adoption, addressing a critical gap between model performance and practical deployment.

The modular architecture facilitated experimentation with different components. For instance, we could independently evaluate the contribution of focal loss, hard negative mining, and temperature scaling. This modularity also aids future development, as components can be upgraded or replaced without redesigning the entire system. The comprehensive evaluation across multiple metrics (ROC-AUC, PR-AUC, learning curves, feature importance) provided a thorough understanding of model behavior beyond simple accuracy.

Challenges: Defining ground-truth complementarity without actual collaboration outcomes required assumptions. Our complementarity score and 75th percentile threshold remain somewhat heuristic, validated only through domain knowledge and empirical performance rather than actual learning outcome data. Obtaining real collaboration data would require longitudinal studies with controlled experiments, presenting ethical and logistical challenges.

Computational constraints limited architecture exploration. Training on CPU rather than GPU extended development cycles and prevented experimentation with deeper architectures or larger

graphs. Memory constraints limited batch sizes and graph sizes, potentially impacting model capacity. Class imbalance remained challenging despite focal loss and hard negative mining, requiring careful monitoring of precision-recall tradeoffs throughout development.

What We'd Change: With more resources, we would partner with instructors to collect validation data measuring actual learning outcomes from recommended pairs. This would enable direct optimization of pedagogical objectives rather than proxy metrics. We would systematically evaluate graph construction choices (varying k, similarity metrics, edge weighting schemes) through ablation studies. GPU access would enable exploration of deeper architectures, larger embeddings, and more sophisticated attention mechanisms. Additionally, we would implement A/B testing infrastructure to evaluate system impact in real educational settings, providing evidence for deployment decisions.

References

[1] Webb, N. M. (1982). *Student interaction and learning in small groups*. Review of Educational Research, 52(3), 421-445.

[2] Kuzilek, J., Hlosta, M., & Zdrahal, Z. (2017). *Open University Learning Analytics dataset*. Scientific Data, 4, 170171.

[3] Nakagawa, H., Iwasawa, Y., & Matsuo, Y. (2019). *Graph-based knowledge tracing*. IEEE/WIC/ACM International Conference on Web Intelligence, 156-163.

[4] Lin, T. Y., et al. (2017). *Focal loss for dense object detection*. IEEE ICCV, 2980-2988.

[5] Guo, C., et al. (2017). *On calibration of modern neural networks*. ICML, 1321-1330.