When Can We Trust AI Coding of Student-Generated Text? A Committee-Based Approach to Diagnosing Agreement and Uncertainty at Scale

Fanjie Li[1 [0000−0001−7016−6354]], Madison Lee Mason[1 [0000−0001−6395−0976]], Daniel T. Levin[1 [0000−0002−2652−0472]], and Alyssa Friend Wise[1 [0000−0002−4043−6808]]

Vanderbilt University, Nashville, TN, USA
{fanjie.li,madison.j.lee,daniel.t.levin,alyssa.wise}@vanderbilt.edu

Abstract. This paper operationalizes a committee-based performance diagnostic framework that combines inter-model agreement, consensus entropy, and borderline rate to support interpretable monitoring of AI coding of student text on unlabeled data. In a pilot application to nursing simulation reflections, these complementary metrics revealed distinct ensemble patterns, including stable consensus and divergence between agreement and decisiveness. The results illustrate how committee diagnostics can support ongoing oversight of AI coding as systems encounter new learners, contexts, and language use at scale.

Keywords: Automated Coding · LLMs · Uncertainty Quantification

1 Introduction

AIED systems increasingly rely on automated interpretation of student-generated text such as written explanations, short responses, and dialogue to deliver adaptive support [2, 11]. Recent advances in large language models have lowered the barriers to large-scale automated text analysis; but the value of such systems depends on maintaining coherent behavior beyond training and testing as models encounter new learners, contexts and forms of expression. In practice, however, such AI coding pipelines rarely include systematic mechanisms for ongoing post-deployment validation or monitoring. Even when methods such as confidence monitoring or drift detection [4] are employed, these techniques typically focus on changes in individual predictions or input distributions, offering limited visibility into whether models continue to produce consistent, aligned interpretations of target constructs once deployed. In educational settings, this can result in misleading feedback, inappropriate agent responses, or misaligned system actions. As the use of LLM-based coding rapidly expands, ensuring that AIED systems continue to respond appropriately to the constructs they are intended to detect across learners, contexts, and time requires new approaches to monitoring automated AI analysis of student-generated text throughout the system lifecycle. To

address this gap, this paper introduces a committee-based diagnostic framework that integrates measures of inter-model agreement, decisiveness, and certainty to support interpretable monitoring of automated coding on unlabeled data. The contribution lies not in the individual metrics themselves, which build on established approaches, but in how they are used together to characterize ensemble behavior for each construct and support actionable interpretation over time. We first introduce the *Reflect* system as a motivating context in nursing education, then describe the proposed approach and diagnostics, and finally demonstrate feasibility through an initial proof-of-concept using pilot data.

2 Background: Nursing Simulations and *Reflect* System

Simulation-based education is central to nursing preparation, giving students realistic clinical practice while cultivating the cognitive and metacognitive work required for clinical judgment [7, 15]. Currently, much of the structured opportunity for guided reflection happens in post-simulation debriefings, but they are often constrained by time, instructor variability, and group dynamics [3]. Structured individual written reflection, especially when anchored in student-identified meaningful events, can extend debriefing by prompting deeper sense-making about actions and reasoning. In practice, however, these reflections are rarely reviewed beyond learner themselves. This represents a missed opportunity to support reflective skills, a core component of professional learning that rarely develops without structured support; and to provide instructional teams insight into students' clinical judgment development across contexts and time.

Reflect is a post-simulation reflection system designed to support nursing students' analysis of their simulation experiences [9]. It was developed as part of a larger project that leverages AI-powered tools to support experiential learning in nursing simulation. *Reflect* promotes sustained re-engagement with the simulation experience by having students segment first-person video into meaningful events and identify associated actions, goals, and cognitive states. The broader project's long-term goals include developing reflection analytics to support adaptive feedback, instructional insight, and indicators of competency development.

While standard test set evaluation of automated reflection coding helps establish baseline performance, it cannot guarantee stability as the system encounters new cohorts and simulation contexts. Without mechanisms for ongoing monitoring, automated coding may drift or produce misleading signals in consequential ways. To address this challenge, we developed a committee-based diagnostic framework that can characterize model behavior on unlabeled data over time.

3 AI Coding Approach and Monitoring Diagnostics

The elements described below augment a standard LLM-based coding pipeline [2, 10] using a committee based approach, with mechanisms to surface uncertainty at three levels: within individual judgments of code presence (via guided chain-of-thought [1] with explicit uncertainty flagging), across each AI coder's

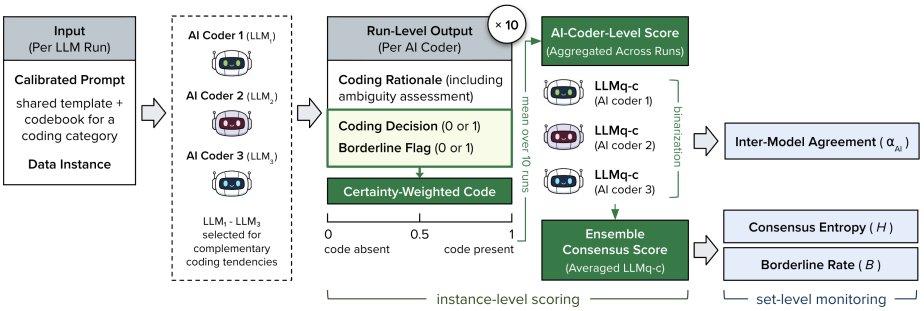

Fig. 1. Committee-based diagnostic framework for performance monitoring.

repeated inference attempts (via decision stability quantification [10, 14]), and among multiple LLMs, each acting as an "AI coder" (via committee agreement [17]). Building on components of our prior active learning framework (Sections 3.1-3.2), this study extends the approach with diagnostic metrics that support ongoing oversight on unlabeled data (Section 3.3; see Fig 1). Below, we describe the approach in its simplest form: binary detection of a single construct on unlabeled data using AI coders calibrated through active learning on labeled training and test sets [8]. Although we present the full suite of techniques here, the approach is modular, allowing adoption of components as appropriate.

3.1 AI Coding System Elements

Guided Reasoning with Uncertainty Flags. When LLMs are used for automated coding, they are typically prompted to produce label predictions with little visibility into decision confidence. Prior work has shown that chain-of-thought prompting, where models articulate step-by-step reasoning before reaching conclusions, can improve both performance and interpretability [2]. We extend this approach by embedding explicit uncertainty signaling within the reasoning process. Specifically, the AI coder is prompted to: (1) reason through application of codebook to a data instance, (2) note moments of decision difficulty (e.g., competing interpretations) and (3) flag cases where code presence or absence cannot be determined with confidence. The output includes both a binary judgment (code present/absent) and an explicit uncertainty flag (raised or not), which together can be used to derive an uncertainty-adjusted score (Section 3.2).

Repeated Inference for Decision Stability. Beyond single-shot inference, each AI coder processes every data item through multiple independent runs (e.g., N=10) with a non-zero temperature (e.g., temperature = 1), following prior work using repeated inference to examine LLM decision stability [10, 14]. This approach draws on the self-consistency hypothesis [16], which posits that, when a construct is well operationalized, multiple valid reasoning paths should converge on the same coding decision. In contrast, variability across repeated inferences signals intra-coder decision instability, often reflecting latent ambiguity or competing interpretations that warrant closer human inspection.

Committee-Based Coding with Multiple LLMs. As a single AI coder's uncertainty estimate may reflect model-specific biases or overconfidence, the system employs a committee of AI coders, each using a different LLM. This ensemble approach follows Query by Committee methods from active learning [12], which use disagreement among diverse models to identify ambiguous cases near the decision boundary. When multiple models independently flag uncertainty or reach different coding decisions, the ensemble provides more robust signals of collective uncertainty not subject to a particular model's training biases.

3.2 Instance-Level Scoring

For each data instance, run-level outputs are aggregated within each AI coder to produce an AI-coder-level score ($LLMq{-}c \in [0, 1]$ described below) summarizing decision stability and uncertainty across stochastic runs. The resulting set of AI-coder-level scores is then used to derive committee-level decisions, and compute diagnostic measures of agreement and decisiveness across the dataset.

Certainty-Weighted LLMq (*LLMq-c*). At the individual coder level, we build on Tai et al. [14]'s Large Language Model quotient (LLMq). LLMq measures model confidence in code presence by averaging binary codes across repeated LLM queries and has recently been used in learning analytics to characterize model behavior across multiple coding runs [10]. Our certainty-weighted variant extends LLMq by down-weighting uncertainty-flagged outputs to the neutral midpoint (0.5), then averaging these uncertainty-adjusted scores across runs (e.g. if 5 LLM runs yield outputs $\{1,1,1,1,1_{flag}\}$, then *LLMq*=1, *LLMq-c*=0.9). This weighting scheme treats uncertainty-flagged codes as providing no directional information, preventing hesitant decisions from artificially inflating or deflating the resulting *LLMq-c* score.

Ensemble Consensus Score ($\hat{p}$). Given prior work demonstrating improved robustness through ensemble aggregation [4, 17], we compute the committee's collective judgment as mean of three AI coders' *LLMq-c* scores; applying a 0.5 threshold produces the final binary decision for downstream analysis.

3.3 Performance Monitoring Metrics

On unlabeled data, traditional validation metrics (e.g., F1) are unavailable. However, the committee-based, uncertainty-aware framework allows system health to be diagnosed through three complementary metrics that require no ground truth labels.

Inter-Model Agreement (α_{AI}). Krippendorff's α [6], a robust and widely used measure of inter-rater reliability, is used to assess inter-model agreement. Based on the three models' thresholded decisions derived from binarized *LLMq-c*, this metric indexes how often the model committee converges on the same coding decision (α>0.67 indicates moderate agreement; α>0.80 strong agreement [6]). Sustained high agreement indicates consistent codebook application across models, while declining α_{AI} signals potential drift, suggesting models beginning to diverge in how they interpret the codebook when applied to new data.

Consensus Entropy (H). This metric is computed as the Shannon entropy $H \in [0, 1]$ over the instance-level, cross-model consensus score ($\hat{p}$) [15], where: $H = -[\hat{p} \times \log_2(\hat{p}) + (1 - \hat{p}) \times \log_2(1 - \hat{p})]$. Following [17], average consensus entropy across items serves as a summary indicator of ensemble decisiveness on unlabeled data. Lower H indicates more decisive ensemble judgments ($\hat{p}$ near 0 or 1); higher H indicates greater indecisiveness ($\hat{p}$ near 0.5) which can occur when models confidently disagree or are uncertain or unstable across runs.

Borderline Rate (B). While consensus entropy summarizes overall decisiveness, it does not distinguish between pervasive uncertainty and uncertainty concentrated on a few items. Building on prior work [5], we distinguish between broad-based uncertainty and uncertainty localized to a subset of ambiguous instances using borderline rate (B), the percentage of items for which $L \leq \hat{p} \leq U$, where L and U are probability bounds defining a region of decisional ambiguity.

Summary of Contribution. While existing studies have explored guided reasoning [1], coding stability scoring [10, 14], and ensemble methods [17] independently, our contribution lies in their novel integration to enable performance monitoring on unlabeled data. Building on our prior work on calibrating AI coders via active learning [8], the present study repurposes consensus entropy (H) and extends the set of diagnostics with borderline rate (B) and inter-model agreement (α_{AI}), using all three metrics for deployment-time monitoring of potential degradation in coding quality. Rather than treating these metrics as traditional measures of reliability or validity, we interpret them as complementary diagnostics of agreement, decisiveness, and certainty, relative to patterns established during calibration and testing. These diagnostics can be continuously monitored in settings where ground truth labels are unavailable or themselves subject to uncertainty [13]. Interpreted together, they provide actionable insight into which codes are likely to be applied robustly across models and repeated inferences and for which further investigation is warranted.

4 Proof-of-Concept Pilot

4.1 Context, Data and Coding Scheme

This pilot uses an event-level Reflect corpus from a nursing school at a U.S. research university (Summer 2024) of 213 reflections from 16 students reflecting on key moments from a range of adult and pediatric simulations. The project's first coding scheme was a simulation-general activity framework aligned with models of clinical judgement [15] and self-regulation [7]. By surfacing behavioral and cognitive elements not fully visible in simulation video, reflections provide a useful basis for interpreting student engagement with key dimensions of practice. The nine-activity coding scheme (Table 1) was iteratively developed through inductive analysis and deductive refinement grounded in nursing education and reflective learning theory [15, 7], informed by consultation with nursing education experts. The resulting codebook includes code definitions, indicators, and decision heuristics; full details are reported elsewhere [8].

Table 1. Simulation-general nursing activity codes for student reflections.

Code	Code Name	Definition
GATHER	Gathering information	Collecting or receiving clinical data related to the patient needed for safe and effective nursing care.
IMPLMT	Implementation of nursing care	Hands-on execution of direct care procedures to address patient needs or problems.
T-COMM	Communication with healthcare team	Communicating or collaborating with healthcare team members to share info or coordinate care.
P-COMM	Communication with patient (family)	Communicating with patients or families to build rapport, share info and address questions.
INTRPT-SIT	Interpret info to understand patient situation	Connecting clinical info to form understanding, hypotheses, or clinical judgments about a patient.
INTRPT-VLDT	Interpret info to validate care plan	Ensuring correctness, safety, and readiness of a planned nursing intervention for implementation.
INTRPT-EVAL	Interpret info to evaluate outcome	Assessing a patient's response to nursing care to determine therapeutic or adverse effects.
GOALS	Establish goals and generate solutions	Planning nursing care by identifying therapeutic goals and appropriate interventions.
TIME	Prioritization and time management	Managing time, tasks, and/or priorities during care delivery.

The codebook was used by three researchers to double code 60 reflections (IRR: $0.63 < \alpha < 0.91$). Following reconciliation, data was split into training (N=36) and test sets (N=24). A relatively large test set was used to ensure sufficient positive instances for a rare category (INTRP-EVAL). Three AI coders (o3, Claude-Sonnet-4, DeepSeek-r1) were calibrated through iterative active learning using a shared system prompt with dynamically inserted relevant codebook sections. These models were selected based on observed differences in coding tendencies during calibration (e.g., more conservative vs. more inclusive labeling), providing complementary perspectives for committee-based assessment. Data instances with high uncertainty triggered targeted human review and prompt refinement until human-AI agreement stabilized, after which performance was evaluated on held-out data. Full prompt engineering and training workflow details are reported in [8]. Here we characterize model behavior on both the test set and unlabeled data using the three diagnostic metrics introduced above: inter-model agreement (α_{AI}), consensus entropy (H), and a borderline rate (B) of $0.35 \leq \hat{p} \leq 0.65$. Bounds were chosen based on patterns in uncertainty observed during calibration. The goal is to provide a proof-of-concept demonstrating how complementary diagnostic signals can be jointly interpreted to characterize ensemble behavior in the absence of ground truth (see Fig 2).

4.2 Results

Table 2 summarizes diagnostic indicators and test-set performance metrics by activity code, comparing agreement and uncertainty patterns across constructs

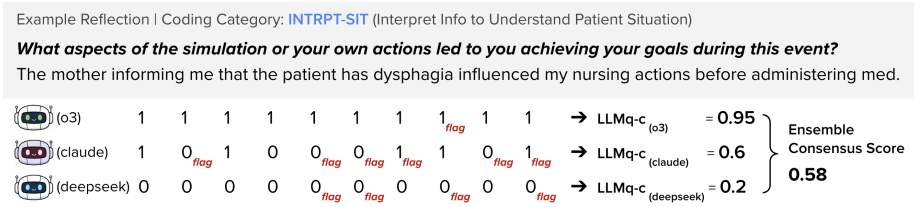

Fig. 2. Example of committee disagreement due to differing expectations for explicit versus implicit evidence of interpretation work (recognizing implications of dysphagia).

Table 2. Performance metrics for test data and unlabeled data.

	Test Data				Unlabeled Data		
Activity Code	F1	α_{AI}	H	B	α_{AI}	H	B
GATHER	0.966	0.942	0.295	4.17%	0.885	0.275	7.69%
IMPLMT	0.941	1.000	0.322	0.00%	0.951	0.218	5.13%
T-COMM	0.923	0.877	0.295	8.33%	0.946	0.230	2.56%
P-COMM	0.923	0.944	0.214	8.33%	0.917	0.131	3.42%
INTRPT-SIT	0.770	0.683	0.344	8.33%	0.756	0.276	9.40%
INTRPT-VLDT	0.947	1.000	0.321	0.00%	0.837	0.343	9.40%
INTRPT-EVAL	1.000	0.844	0.195	0.00%	0.532	0.163	1.71%
GOALS	0.824	0.698	0.506	25.00%	0.719	0.456	12.82%
TIME	1.000	0.735	0.379	4.17%	0.644	0.309	11.11%

for both held-out test and unlabeled data. For the first four behaviorally explicit codes (GATHER, IMPLMT, T-COMM, P-COMM) with very high test-set performance (F1 $>$ .9), the ensemble showed stable high inter-model agreement across test and unlabeled data (all $\alpha_{AI} > .8$). Consensus entropy remained relatively stable and borderline rates stayed below 10% with minimal changes. This stable-consensus profile is consistent with expectations for behaviorally explicit constructs and suggests diagnostically stable generalization beyond held-out evaluation. While very high test-set F1 scores raise overfitting concerns, diagnostics on unlabeled corpus provide complementary evidence that ensemble behavior remained stable rather than brittle on additional data.

The latter five codes require more inference about cognitive activity, yielding a broader range of diagnostically informative patterns. INTRP-SIT showed lower test-set performance (F1=0.77) than behaviorally explicit codes, consistent with its more interpretive nature (Fig 2). However, inter-model agreement, entropy, and borderline rates remained stable across test and unlabeled data (α_{AI}=0.683/0.756, H=0.344/0.276, B=8.33%/9.40%), suggesting that the lower accuracy did not translate into unstable ensemble behavior when applied to additional unlabeled data. In contrast, INTRP-VLDT showed perfect test set agreement (α_{AI}=1.0) paired with modest ensemble entropy (H=0.321) but no borderline cases, indicating unanimous but weakly decisive judgments. On unlabeled data, a rise in borderline cases without a corresponding increase in entropy, together with an expected decline in inter-model agreement, suggests uncertainty

concentrated in a subset of cases. As validating a nursing care plan requires attention to safety-critical practices varying by clinical contexts, this pattern is consistent with context-sensitive variation in how validation reasoning is expressed and points to a need for targeted human review of emerging edge cases.

INTRP-EVAL was a rare code in the dataset, making its perfect test-set accuracy (F1=1.0) potentially misleading. On unlabeled data, α_{AI} dropped substantially ($0.884 \rightarrow 0.532$), while entropy and borderline rates remained low ($H < 0.2$, $B < 2\%$). This pattern reflects confident consensus on code absence for most items, paired with disagreement on the small number of candidate positive cases. In large unlabeled datasets, this pattern can be used to flag a small subset of instances where an AI coder indicates potential code presence, enabling targeted human review.

GOALS showed good performance on the test set (F1=0.824) with moderate agreement (α_{AI}=0.698/0.719) but midrange entropy (H=0.506/0.456) and elevated borderline rates (B=25%/13%) indicating pervasive uncertainty across test and unlabeled data. This pattern suggests potential ambiguity in how the GOALS construct is operationalized. Consistent with challenges in human coding, this reflects a tension between a theoretically grounded GOALS construct and the partial or emergent ways novice learners express goal setting and solution generation. This points to a need to refine code definitions or prompt guidance, rather than concerns about model stability when applied to new data.

Finally, TIME also showed perfect test-set accuracy (F1=1.0), with only moderate inter-model agreement (α_{AI}=0.735). On unlabeled data, agreement declined further (α_{AI}=0.644) while borderline rate rose (4%→11%) without a corresponding rise in entropy, indicating increased uncertainty in a subset of cases. This pattern of declining agreement and increased localized uncertainty suggests difficulty distinguishing *prioritization and time management* (TIME) from adjacent clinical reasoning, consistent with the construct's cognitive complexity and indicating a need for further construct clarification or training data.

5 Limitations and Conclusion

Using a committee-based framework, this paper operationalizes complementary metrics for continuous monitoring of AI coding performance on unlabeled data. In a pilot application, these diagnostics revealed recurring ensemble patterns, including stable consensus and divergence between agreement and decisiveness, illustrating how the metrics can be jointly interpreted to monitor AI coding in the absence of ground truth. Limitations include the small pilot scale and its focus on reflection data from a single nursing program using one coding framework. Further validation is needed to assess whether the metrics function as useful diagnostics across other forms of student-generated text, coding schemes, and instructional contexts. The computational cost of committee-based inference is also non-trivial; future work will explore sparse and adaptive monitoring strategies to balance efficiency with ongoing oversight. Overall, this work addresses an important challenge in AIED: maintaining the robustness of AI coding of com-

plex, open-ended student work as systems encounter new populations, contexts, and patterns of expression at scale.

Acknowledgments. This work was supported by the National Science Foundation under Grant No. DRL-2418602. The opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

1. Anthropic: Let Claude think (CoT), https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/chain-of-thought
2. Cohn, C., Hutchins, N., Le, T., Biswas, G.: A chain-of-thought prompting approach with LLMs for evaluating students' formative assessment responses in science. In: Proc. AAAI Conference on Artificial Intelligence. pp. 23182–23190 (2024)
3. Decker, S., et al.: Standards of best practice: Simulation standard VI: The debriefing process. Clinical Simulation in Nursing **9**(6), 26–29 (2013)
4. Herrera-Poyatos, D., et al.: An overview of model uncertainty and variability in LLM-based sentiment analysis: challenges, mitigation strategies, and the role of explainability. Frontiers in Artificial Intelligence **8**, 1–24 (2025)
5. Jamison, E., Gurevych, I.: Noise or additional information? leveraging crowdsource annotation item agreement for natural language tasks. In: Proc. 2015 Conference on Empirical Methods in Natural Language Processing. pp. 291–297 (2015)
6. Krippendorff, K.: Content analysis: An introduction to its methodology (2019)
7. Lajoie, S.P., Gube, M.: Adaptive expertise in medical education: Accelerating learning trajectories by fostering self-regulated learning. Medical Teacher **40**(8) (2018)
8. Li, F., Mason, M.L., Levin, D., Wise, A.: Using RE-LLM coding uncertainty to resolve codebook ambiguities: An example of the CLARIFY toolset and workflow in action. In: Joint Proceedings of LAK 2026 Workshops. pp. 1–10 (2026)
9. Mason, M.L., et al.: Transforming experiences into expertise: Leveraging event cognition to support self-regulation in a practical learning system (2026), in review
10. Ramanathan, S., et al.: When the prompt becomes the codebook: Grounded prompt engineering (GROPROE) and its application to belonging analytics. In: LAK'25 Proceedings. pp. 713–725. ACM (2025)
11. Scarlatos, A., Baker, R.S., Lan, A.: Exploring knowledge tracing in tutor-student dialogues using LLMs. In: LAK'25 Proceedings. pp. 249–259. ACM (2025)
12. Settles, B.: Active learning literature survey. Tech. Rep. TR1648, University of Wisconsin-Madison Department of Computer Sciences (2009)
13. Song, H., et al.: In validations we trust? the impact of imperfect human annotations as a gold standard on the quality of validation of automated content analysis. Political Communication **37**(4), 550–572 (2020)
14. Tai, R.H., et al.: An examination of the use of large language models to aid analysis of textual data. International Journal of Qualitative Methods **23**, 1–14 (2024)
15. Tanner, C.A.: Thinking like a nurse: A research-based model of clinical judgment in nursing. Journal of Nursing Education **45**(6), 204–211 (2006)
16. Wang, X., et al.: Self-consistency improves chain of thought reasoning in language models (2023), arXiv:2203.11171 [cs.CL]
17. Zhang, Y., et al.: Consensus entropy: Harnessing multi-vlm agreement for self-verifying and self-improving OCR (2025), arXiv:2203.11171 [cs.CV]