The Illusion of Progress: Dialogue-Derived Signals for Proactive Intervention Timing in Student–GenAI Interaction

Fangwei Lin[1,*], Khe Foon Hew[1]

[1]*Faculty of Education, The University of Hong Kong, Hong Kong SAR, China*

Abstract

When should a GenAI assistant intervene, and when should it remain silent? We argue that the hardest timing problem is not detecting inactivity, but recognizing the "illusion of progress": interaction that appears productive without translating into applied action such as executing code or testing solutions. Learners may continuously ask questions and build conceptual understanding without ever acting on that knowledge. In dialogue-centric settings, interaction logs contain the behavioral traces needed to detect this understanding–enactment gap. Drawing on 2,243 dialogue turns from 136 undergraduates in a data mining course, we identify three kinds of timing-relevant signals: stalling patterns, enactment cues, and the integration trap, a state in which understanding advances without resolution. Together, these signals make the gap visible. We therefore argue for a restraint-first intervention policy: systems should intervene not merely because dialogue appears active, but when sustained engagement fails to convert into observable enactment, while withholding intervention once concrete enactment cues are present.

Keywords

Dialogue Logs, Generative AI, Proactive Intervention Timing, Self-Regulated Learning, Human–AI Interaction

1. Introduction

Generative AI (GenAI) assistants still operate largely in a reactive mode, responding when prompted and remaining silent otherwise [1]. In learning settings, however, timing matters. Decades of tutoring research show that whether feedback helps depends partly on when it arrives [2, 3], yet the question of when a GenAI system should intervene proactively remains underexplored [4, 5].

Some timing cases are relatively easy to interpret. A student who repeatedly issues low-level queries is visibly stuck, whereas a student submitting code, outputs, or debugging traces is already showing evidence of enactment. The harder case is the learner whose dialogue appears engaged and purposeful but never translates into action. Such learners may ask progressively sophisticated questions, connect ideas, and articulate plausible explanations, yet never actually test, apply, or verify anything. From the outside, this can look like progress. In practice, it may be a more elaborate form of stalling. We refer to this phenomenon generally as the *illusion of progress*; theoretically, we interpret it as a form of *regulatory misalignment*; and empirically, it surfaces in dialogue logs as the *integration trap*.

This distinction matters because action is where self-regulated learning actually operates. In cyclical SRL models, learners set goals, execute strategies, observe outcomes, and adjust accordingly [6, 7]. Without executed action, there are no outcomes to observe and no feedback to regulate against. Understanding that remains unenacted is what Whitehead [8] termed *inert ideas*: received into the mind but never utilised, tested, or thrown into fresh combinations.

Dialogue logs are especially well-suited to detecting this gap between understanding and action. Physiological sensing can suggest whether a learner is aroused or attentive [9], but it does not directly indicate whether the learner is moving from understanding to enacted action. In dialogue-centric

Proceedings of TimelyAI: When Should Generative AI Assistants Intervene? A Workshop Co-located with the 5th Symposium on Human-Computer Interaction for Work (CHIWORK 2026), Linz, Austria, June 22 to 25, 2026

*Corresponding author.

fangwei.lin@connect.hku.hk (F. Lin); kfhew@hku.hk (K. F. Hew)

0009-0005-1959-7876 (F. Lin)

settings, the interaction log already captures the most relevant evidence for that distinction, because it records goals, requests, outputs, and the progression of inquiry as a byproduct of normal interaction.

Student–GenAI interaction offers a tractable setting for making this argument. In adaptive learning research, the cost of mistimed support is well established, from maladaptive help-seeking in intelligent tutoring systems [10] to the failure of fixed scaffolding that ignores learners' regulatory states [11]. At the same time, interactions unfold across multiple turns, producing dialogue logs that make temporal patterns visible. More importantly, this understanding–enactment gap often surfaces directly in the log. The remainder of the paper uses this visibility. §2 frames the timing problem as regulatory misalignment. §3 identifies detectable signal patterns in dialogue. §5 proposes a cumulative threshold model for multi-tiered, restraint-first intervention.

2. Empirical Foundation

This paper analyzes 2,243 student-initiated dialogue turns from 136 undergraduates who interacted with a GenAI assistant over four weeks in a data mining course, where AI use was explicitly encouraged for both conceptual learning and programming.[1] The course was heavily project-based, requiring students to actively write and execute analysis code. Final course grades therefore strongly reflect the enactment behaviors captured in the chat logs, and we use these grades to distinguish high from low achievers. In this technical context, turns that involved conceptual inquiry (e.g., asking about algorithms) and turns that involved hands-on implementation (e.g., submitting code or outputs) were systematically distinguished in the log, providing the empirical basis for our timing framework.

Our analysis adopts a control-theoretic view of self-regulated learning (SRL), in which productive learning depends on coordinating goals, actions, and monitored progress over time [12, 13, 6]. We operationalize these three regulatory functions through *Learning Intention* (LI), *Prompt Communicative Function* (PCF), and *Cognitive Presence* (CP) (see Table 1). LI captures the learner's current cognitive goal; PCF captures the interactional move directed at the GenAI assistant (e.g., requesting information vs. submitting one's own output); and CP tracks how far the inquiry has progressed moving from problem identification toward resolution.

For intervention timing, no single dimension is sufficient. Intent is not the same as action, interactional moves have different meanings at different phases, and apparent inquiry progress does not guarantee enactment. What matters is whether these dimensions remain aligned as the learner moves toward action. Checking this alignment is complicated because each dimension carries its diagnostic information in a different region. For LI, the relevant question is at the low end of the hierarchy: whether a learner remains cycling between Remember and Understand, or transitions upward. High achievers make this upward transition at roughly three times the rate of low achievers. For CP, by contrast, the low end carries almost no diagnostic value: 75.1% of all coded turns remain at Triggering regardless of achievement level. Discrimination emerges only at the rare upper phases, where lower achievers are systematically displaced toward Integration (Cohen's $d = 0.55$) without reaching Resolution.

This alignment logic makes the central problem in this paper visible. Apparent progress without enactment is a case of *regulatory misalignment*, in which the learner's goals may grow more ambitious and the inquiry may advance, while action remains uncommitted. In this dataset, that pattern appeared most clearly in the *integration trap*, in which understanding consolidated without reaching resolution. We foreground this trap because, unlike low-level stalling, it is indistinguishable from genuine deep learning on the dialogue surface alone; only the persistent absence of enactment signals reveals it. Table 2 illustrates this contrast with a simplified scenario. The following section identifies the dialogue patterns that make such states visible for intervention timing.

[1] The empirical patterns reinterpreted here derive from a larger unpublished study currently under review; the present paper summarizes only the findings needed to motivate the timing framework.

Table 1
Coding dimensions and definitions. Signal relevance for intervention timing (stalling, non-diagnostic, or enactment) is shown in Figure 1.

Dimension	Code	Definition
Learning Intention	Remember/Understand	Recall or comprehend concepts
	Apply/Analyze	Use or examine knowledge
	Evaluate/Create	Judge, critique, or produce
Prompt Comm. Function	Knowledge Query (KQ)	Ask a factual question
	Contextualized Query (CQ)	Ask with situational context
	Task Delegation (TD)	Request AI to perform a task
	Elaboration (EL)	Request expansion or detail
	Output Shaping (OS)	Modify AI-generated output
	Artifact Submission (AS)	Submit own execution results
Cognitive Presence	Triggering/Exploration	Identify problem; gather info
	Integration	Synthesize into understanding
	Resolution	Apply or test a solution

Table 2
Misaligned vs. aligned interaction in a data mining course. Student A asks increasingly deep questions (CP advances to Integration) but never acts; Student B reaches Resolution in three turns. Colored cells mark diagnostic differences (red for the integration trap, green for enactment).

Turn	Student utterance (paraphrased)	LI	PCF	CP
Student A — Misaligned (integration trap)				
1	"How is information gain calculated?"	Understand	KQ	Triggering
2	"How does Gini differ from info gain?"	Understand	KQ	Exploration
3	"What about continuous features?"	Understand	KQ	Exploration
4	"So Gini is better for continuous features?"	Understand	KQ	Integration
5	"Can I mix both? What does sklearn default to?"	**Understand**	**KQ**	**Integration**
...	*(Continues conceptual querying without execution)*	**...**	**...**	**...**
Student B — Aligned (enacted)				
1	"How is information gain calculated?"	Understand	KQ	Triggering
2	"Write me a decision tree with sklearn"	Apply	TD	Exploration
3	"I ran it—got `ValueError`; here's my output"	**Apply**	**AS**	**Resolution**

3. Dialogue-Derived Signals for Intervention Timing

Through the timing lens developed above, the empirical patterns yield three classes of dialogue-derived signals. Each class corresponds to a distinct misreading a proactive system can make of a learner's trajectory toward enactment. A system might fail to detect stalling, miss enacted progress, or mistake deep dialogue for actual execution. Figure 1 maps each dimension's codes by their diagnostic relevance to these classes.

3.1. When Interaction Stalls

A stalling signal is not the absence of interaction, but sustained interaction without movement toward enactment (the warm-colored zones and looping arrows in Figure 1). In the data, this pattern appeared in two characteristic forms.

The first is *sustained low-level cycling*. Here, recent prompts remain concentrated in Remember and Understand without progressing toward Apply or Evaluate. Dialogue continues and learners may articulate deeper insights, but action remains uncommitted. As previewed in §2, the absence of upward

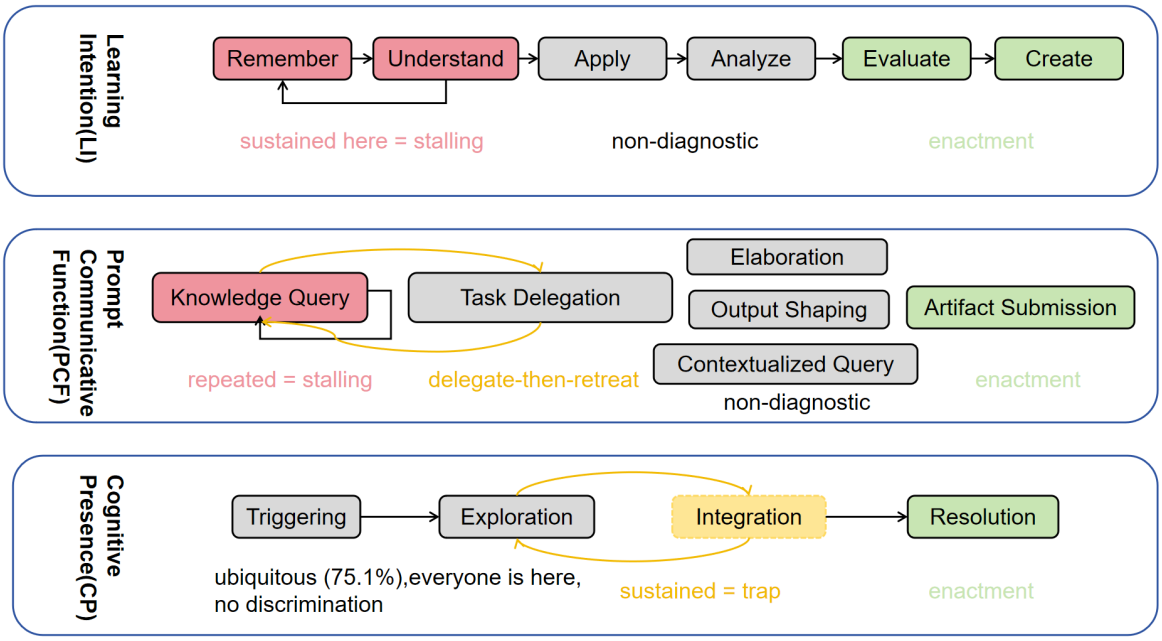

Figure 1: Diagnostic signal structure of three coding dimensions for intervention timing. Each dimension arranges its codes by diagnostic value: codes in the *stalling zone* (warm-colored, left) indicate potential stalling when sustained or repeated; *non-diagnostic* codes (grey, center) carry no timing information; codes in the *enactment zone* (green, right) confirm enacted progress upon a single occurrence. Looping arrows denote characteristic stalling patterns: sustained low-level cycling in LI (Remember↔Understand), repeated querying and delegate-then-retreat cycling in PCF (Knowledge Query ↔ Task Delegation), and the Exploration↔Integration loop in CP that constitutes the Integration Trap. Note that LI and CP follow ordered progressions (→), whereas PCF codes are unordered functional categories. CP's early phases (Triggering, Exploration) are grey rather than warm because they are ubiquitous across all learners (75.1% of turns) and therefore non-discriminating for timing.

transitions through Bloom's levels (e.g., Understand→Apply) functions as a reliable stalling signal.

The second is *delegate-then-retreat* (TD→KQ). A student delegates a task to the AI (e.g., "Write the code for...") but, instead of testing or verifying the output, immediately returns to conceptual questioning. This pattern was significantly more frequent among lower-achieving students, indicating that AI-generated output did not lead to enacted follow-through.

3.2. When Interaction Moves Toward Enactment

Where stalling means sustained interaction without action, enactment is its converse. Learners begin testing, executing, or verifying AI-generated outputs. The green zones in Figure 1 mark the codes that confirm such enacted progress.

The most diagnostic enactment signal is *Artifact Submission* (AS): turns in which learners return with execution results such as error messages, code outputs, or diagnostic traces. AS provides *learner-generated externalized evidence of enactment*. In the underlying study, it showed the largest group difference across all codes, with high achievers submitting artifacts at nearly three times the rate of low achievers.

Equally telling are *delegate-then-verify* sequences (TD→AS), in which a learner delegates a task and then returns with execution results. This close-the-loop pattern occurred at roughly four times the rate in high achievers compared to low achievers. Sustained debugging chains (AS→AS→AS) were present in high achievers but entirely absent in low achievers. This indicates that iterative enactment is itself a learnable interactional skill that timing systems may be able to scaffold.

For timing, these patterns provide the clearest evidence that the learner is already acting. Interrupting here risks disrupting exactly the self-regulated movement the system should preserve.

3.3. When Understanding Advances but Action Does Not

The markers above operate at the scale of individual turns. The most difficult timing errors, however, emerge at a broader temporal scale, when inquiry advances in the dialogue without converting into action.

In the underlying study, this pattern manifested as the *integration trap* (the Exploration↔Integration loop in Figure 1). Learners reached the Integration phase, connecting ideas into coherent understanding, but chronically failed to progress to Resolution, where a solution is applied or tested. As noted in §2, network analysis [14] confirmed this systematic displacement toward Integration, with Integration accounting for only 3.2% and Resolution for 4.4% of coded turns. Because the interaction looks thoughtful and progressive rather than idle, a system monitoring only local cues would easily mistake this state for productive learning.

This pattern may also be reinforced by the assistant's response style. When conceptual prompts are repeatedly met with fluent and expansive explanations, the interaction may continue to feel productive while remaining short of testing, verification, or artifact submission. The integration trap should therefore be treated as an interactional pattern visible in learner dialogue, rather than as evidence of a learner trait.

Crucially, these differences in inquiry progression persisted after controlling for prior academic performance, whereas differences in learning intention did not. The failure to transition into resolution is therefore a specific interactional breakdown, not a proxy for academic ability—and thus a viable target for timing interventions.

Taken together, these three classes of signals converge on a single principle. The relevant question for timing is not whether dialogue is active, but whether the learner's trajectory is advancing toward action. Uncommitted stalling, externally visible enactment, and the integration trap each require a fundamentally different response from a timing-sensitive system.

4. From Signals to Real-Time Detection

Turning these signals into real-time timing decisions means moving from post-hoc analysis to online classification. Intelligent tutoring systems have already made this transition for simpler interaction traces, including off-task behavior [15, 16], maladaptive help-seeking [10], and affective states [17]. Student–GenAI dialogue presents a richer but harder case, since the system must classify natural-language prompts and interpret their trajectory over multiple turns.

Operationalizing the signals from §3 requires real-time classification along the same three dimensions. In the underlying study, two trained coders applied the scheme with acceptable inter-rater reliability [18], confirming the signal categories are consistently recognizable. Production systems must automate this classification step. Recent advances in NLP-based epistemic network analysis [19] and LLM-based qualitative coding now make this operationally feasible. A practical architecture classifies each incoming prompt at the turn level, then monitors these turns through a sliding window to match stalling and enactment patterns. Because these signals derive entirely from text the assistant must parse anyway, the framework requires no additional infrastructure.

5. A Cumulative Threshold Model for Timing Decisions

Each pattern admits innocent explanations. A single TD→KQ bigram may simply reflect clarification, and one session without AS may reflect a careful reader rather than a stalled one. Raw detections must therefore be accumulated rather than acted on individually [20]. We capture this logic in a *cumulative threshold model* (Figure 2).

The model operates over a sliding window of recent turns (in the empirical data, a window of approximately five turns spans a typical question–response–followup cycle). Within each window, the system monitors for four stalling patterns identified in §3, each incrementing a running stalling score:

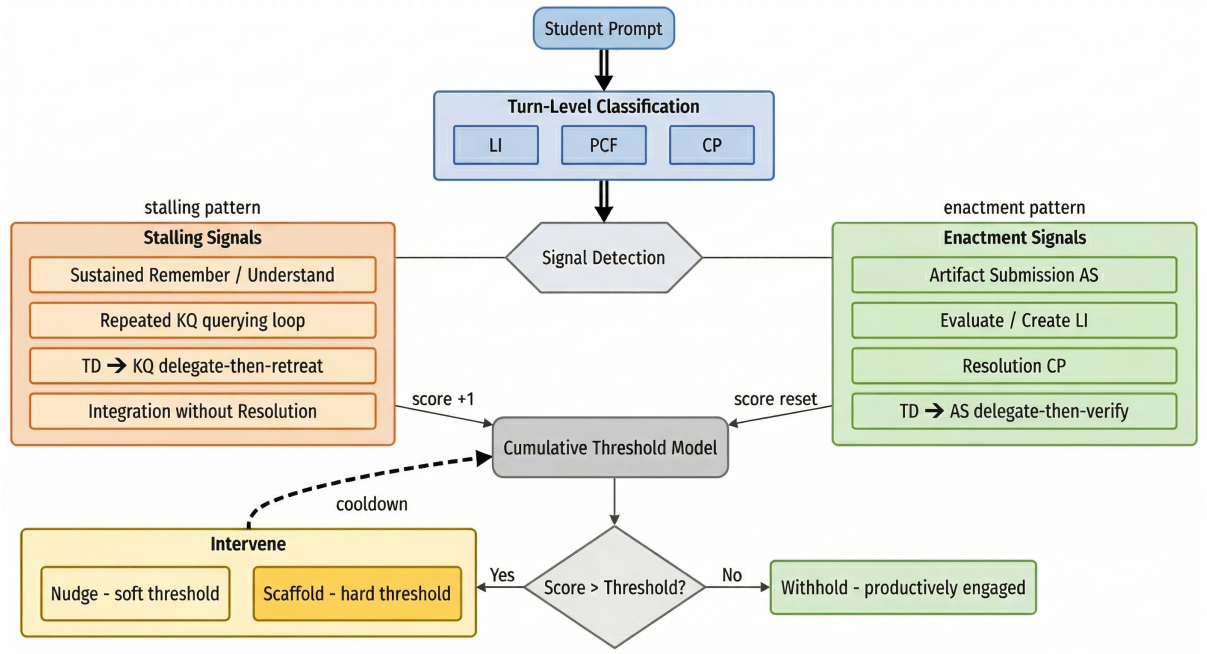

Figure 2: Intervention Timing Pipeline. Student prompts are classified along three dimensions, producing stalling or enactment signals. A cumulative threshold model determines whether to nudge, scaffold, or withhold intervention.

- **Sustained low-level cycling.** Recent turns remain concentrated at Remember/Understand in LI without upward transition toward Apply or Evaluate.
- **Repeated querying.** Knowledge Query (KQ) recurs across consecutive turns without other communicative functions intervening.
- **Delegate-then-retreat.** A Task Delegation (TD) turn is followed by a return to Knowledge Query rather than Artifact Submission, indicating that AI-generated output was not enacted upon.
- **Integration without resolution.** The learner has reached the Integration phase in CP, but no Resolution turn has appeared within the current window.

Each detected pattern increments the stalling score, weighted by inquiry phase. A TD→KQ sequence during the Triggering phase may reflect genuine exploration, while the same pattern during Integration more likely indicates stalling. Conversely, any enactment signal resets the stalling score to zero: Artifact Submission, Evaluate/Create LI, Resolution CP, or a recognized TD→AS sequence each qualify. This asymmetry is deliberate. A single piece of externally visible action outweighs accumulated suspicion.

When the stalling score exceeds a calibrated threshold and no enactment reset has occurred within the current window, the system intervenes. A cooldown period after each intervention suppresses further triggers, reducing the risk of over-scaffolding.

The model supports three multi-tiered response levels, ranging from a *nudge* at a soft threshold (e.g., "Have you tried running this code?"), through a more directive *scaffold* at a harder threshold (e.g., "Try pasting your error message here"), to a *withhold* that suppresses suggestions when enactment signals are already present.

Although the model is described primarily in terms of student prompts, assistant responses are also relevant to timing. Future implementations could extend this learner-side signal model by incorporating features of assistant responses, such as whether the assistant opens additional conceptual branches or instead invites testing, verification, or artifact submission.

Evaluating whether an intervention was *timely*, rather than merely *delivered*, requires metrics at three timescales: *turn-level* transition and phase-progression rates, *session-level* enactment ratios and integration-trap escape rates, and *longitudinal* self-regulation transfer. A controlled experiment

comparing dialogue-timed, random-timed (yoked for frequency), and reactive-only conditions can isolate the effect of timing accuracy from intervention frequency.

6. Discussion

The analysis above yields a design stance that extends beyond the specific model. **A proactive assistant should default to restraint.** This is not because intervention is unhelpful. Rather, deep conceptual reflection and unproductive stalling appear identical until enactment cues emerge. The boundary between contemplation and stall remains the framework's least resolved problem, and this stance has consequences for how its limitations should be understood.

This vulnerability propagates into two downstream concerns. The first is *autonomy cost*. Every intervention, even a correctly timed one, signals to the student that the system "knows better," potentially discouraging self-initiated enactment in future interactions. In educational contexts, where the goal extends beyond task completion to the development of self-regulation capacity [6, 7], a system that reliably nudges students toward enactment inadvertently trains them to *wait for nudges*. At the metacognitive level, this risks creating what Koriat and Bjork [21] term an "illusion of competence": learners mistake AI-assisted fluency for genuine mastery, a downstream consequence distinct from the illusion of progress itself but compounding its effects.

The second is *interaction style diversity*. Students who prefer depth-first exploration (many conceptual questions before any testing) will systematically trigger more false stalling signals than breadth-first iterators. A one-size-fits-all threshold therefore risks over-intervening on learners whose reflective style is most valuable. Personalized baselines, calibrated from each student's early interaction history, could address this issue but introduce cold-start problems of their own.

The empirical patterns informing this framework come from a single-institution, observational study in a technical domain (data mining), and two limitations follow. First, the current evidence is correlational. The data show that high achievers *engage in* more enactment, but do not establish that *scaffolding* enactment causes improvement. Whether intervening at the contemplation–stall boundary actually improves outcomes remains untested. Second, the boundary itself may manifest differently across domains. Artifact Submission, the framework's most diagnostic enactment signal, is most naturally observable in coding contexts. In writing or design, enactment may surface as draft submissions or annotated sketches rather than execution logs. The signal taxonomy is intended to be domain-general in *structure*, but its operationalization must be calibrated to each domain's characteristic artifacts.

A further boundary condition concerns the unit of analysis. The present framework was developed for individual learner–AI interaction, where a learner's dialogue trajectory can be interpreted as that learner's movement toward or away from enactment. In collaborative settings, this interpretation becomes more complex because agency is distributed across participants. A learner who remains in Integration may not be stalled individually, but may be waiting for a teammate to execute, test, or report results. Likewise, Artifact Submission may signal group-level enactment rather than the action of the same learner who initiated the preceding query. Extending the framework to collaborative learning and knowledge-work contexts therefore requires shifting the unit of analysis from individual prompt sequences to group trajectories, including who formulates the problem, who delegates, who acts, who verifies, and how these roles coordinate over time.

Three questions remain open. The first is *unobserved enactment*: a learner might successfully grasp a concept and leave the assistant to apply it in their own environment (e.g., an IDE) without ever returning to log a resolution turn. To a dialogue-only monitor, this successful departure looks identical to the integration trap; future systems may need IDE telemetry to close the monitoring loop.

The second is the *sufficiency* question: is the persistent absence of enactment over N turns enough to trigger intervention? Or must systems also detect affective and motivational cues [17, 22] to distinguish genuine stalling from productive thinking?

The third is the *modality* question: whether log-based timing can ever be adequate on its own, or

whether the contemplation–stall boundary ultimately requires multimodal augmentation.

All three questions point to the same fundamental tension. Dialogue logs are uniquely positioned to detect the gap between understanding and action, but they cannot observe what happens outside the conversation. The ceiling of dialogue-only timing will ultimately determine whether this approach can stand alone or must be embedded in a richer sensing architecture. If dialogue-derived timing proves effective even within these constraints, however, the approach is likely to transfer to other GenAI-assisted knowledge work [23, 24].

7. Conclusion

This paper reframes intervention timing by focusing on interaction that appears productive but stalls short of enactment, representing the hardest case for any proactive system. More broadly, it argues that timing should be grounded not in activity alone, but in whether interaction is moving toward enacted action. Dialogue logs make that distinction visible through signals of stalling, enactment, and the integration trap. The next step is to test whether systems can distinguish reflection from stall reliably enough to support enactment without undermining learner self-regulation.

Acknowledgments

This work was supported by the Research Grants Council of Hong Kong Research Fellow Scheme (Reference no: RFS2223-7H02).

Declaration on Generative AI

Generative AI (e.g., ChatGPT-5.1) was used to improve the readability and clarity of the manuscript. After using this tool, the authors carefully reviewed the content. They fully accept responsibility for the publication's content.

References

[1] C. K. Lo, What is the impact of chatgpt on education? a rapid review of the literature, Education sciences 13 (2023) 410. doi:10.3390/educsci13040410.

[2] V. J. Shute, Focus on formative feedback, Review of Educational Research 78 (2008) 153–189. doi:10.3102/0034654307313795.

[3] K. VanLehn, The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems, Educational psychologist 46 (2011) 197–221. doi:10.1080/00461520.2011.611369.

[4] H. Güner, E. Er, Ai in the classroom: Exploring students' interaction with chatgpt in programming learning, Education and information technologies 30 (2025) 12681–12707. doi:10.1007/s10639-025-13337-7.

[5] D. Sun, A. Boudouaia, J. Yang, J. Xu, Investigating students' programming behaviors, interaction qualities and perceptions through prompt-based learning in ChatGPT, Humanities and Social Sciences Communications 11 (2024) Article 1447. URL: https://doi.org/10.1057/s41599-024-03991-6. doi:10.1057/s41599-024-03991-6.

[6] B. J. Zimmerman, Becoming a self-regulated learner: An overview, Theory into Practice 41 (2002) 64–70. doi:10.1207/s15430421tip4102_2.

[7] P. H. Winne, A. F. Hadwin, Studying as self-regulated learning, in: D. J. Hacker, J. Dunlosky, A. C. Graesser (Eds.), Metacognition in Educational Theory and Practice, Lawrence Erlbaum Associates, Mahwah, NJ, 1998, pp. 277–304.

[8] A. N. Whitehead, The Aims of Education and Other Essays, Macmillan, New York, 1929.

[9] X. Ochoa, M. Worsley, Augmenting learning analytics with multimodal sensory data, Journal of Learning Analytics 3 (2016) 213–219. doi:10.18608/jla.2016.32.10.

[10] I. Roll, V. Aleven, B. M. McLaren, K. R. Koedinger, Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system, Learning and Instruction 21 (2011) 267–280. doi:10.1016/j.learninstruc.2010.07.004.

[11] R. Azevedo, J. G. Cromley, D. Seibert, Does adaptive scaffolding facilitate students' ability to regulate their learning with hypermedia?, Contemporary Educational Psychology 29 (2004) 344–370. doi:10.1016/j.cedpsych.2003.09.002.

[12] C. S. Carver, M. F. Scheier, On the Self-Regulation of Behavior, Cambridge University Press, Cambridge, 1998. doi:10.1017/CBO9781139174794.

[13] P. H. Winne, A metacognitive view of individual differences in self-regulated learning, Learning and Individual Differences 8 (1996) 327–353.

[14] A. Csanadi, B. Eagan, I. Kollar, D. W. Shaffer, F. Fischer, When coding-and-counting is not enough: Using epistemic network analysis (ENA) to analyze verbal data in CSCL research, International Journal of Computer-Supported Collaborative Learning 13 (2018) 419–438. doi:10.1007/s11412-018-9292-z.

[15] R. S. Baker, A. T. Corbett, K. R. Koedinger, A. Z. Wagner, Off-task behavior in the cognitive tutor classroom: When students "game the system", in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2004, pp. 383–390. doi:10.1145/985692.985741.

[16] R. Baker, K. Yacef, The state of educational data mining in 2009: A review and future visions, Journal of Educational Data Mining 1 (2009) 3–17.

[17] S. D'Mello, A. Graesser, Dynamics of affective states during complex learning, Learning and Instruction 22 (2012) 145–157. doi:10.1016/j.learninstruc.2011.10.001.

[18] K. Krippendorff, Content Analysis: An Introduction to Its Methodology, 4th ed., SAGE Publications, Thousand Oaks, CA, 2018.

[19] W. Li, H.-Y. Chang, A. Bradford, L. Gerard, M. C. Linn, Combining natural language processing with epistemic network analysis to investigate student knowledge integration within an ai dialog, Journal of science education and technology 34 (2025) 980–993. doi:10.1007/s10956-024-10176-y.

[20] A. C. Graesser, D. S. McNamara, K. VanLehn, Scaffolding deep comprehension strategies through Point&Query, AutoTutor, and iSTART, Educational Psychologist 40 (2005) 225–234. doi:10.1207/s15326985ep4004_4.

[21] A. Koriat, R. A. Bjork, Illusions of competence in monitoring one's knowledge during study., Journal of Experimental Psychology: Learning, Memory, and Cognition 31 (2005) 187–194. doi:10.1037/0278-7393.31.2.187.

[22] M. V. Covington, Making the Grade: A Self-Worth Perspective on Motivation and School Reform, Cambridge University Press, Cambridge, 1992.

[23] S. Noy, W. Zhang, Experimental evidence on the productivity effects of generative artificial intelligence, Science 381 (2023) 187–192. doi:10.1126/science.adh2586.

[24] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, D. C. Schmidt, A prompt pattern catalog to enhance prompt engineering with chatgpt, in: Proceedings of the 30th Conference on Pattern Languages of Programs, PLoP '23, The Hillside Group, USA, 2023. doi:10.5555/3721041.3721046.