StudySense: A Context-Aware Learning Agent with Event-Based Sensing and Long-Term Memory

Zihao Huang
College of Software, Nankai University
Tianjin, China
2413989@mail.nankai.edu.cn

Tianyi Zhang
Nankai University
Tianjin, China
1563327159@qq.com

Ziyuan Lin
College of Cryptology and Cyber Science, Nankai University
Tianjin, China
2410936@mail.nankai.edu.cn

Yuchen Jin
School of Software Engineering, Nankai University
Tianjin, Tianjin, China
2413411@mail.nankai.edu.cn

Zihan Cai
Nankai University
Tianjin, China
jadeandstone7291@gmail.com

Xiaoteng Pan
College of Software, Nankai University
Tianjin, China
2120240803@mail.nankai.edu.cn

Huize Chao*
Tsinghua University
Beijing, China
chz24@mails.tsinghua.edu.cn

Abstract

Generative AI can answer questions and generate practice, but everyday learning unfolds through connected events: students capture problems, make mistakes, ask follow-up questions, practice, and review. Treating each interaction independently prevents an AI system from recognizing recurring misconceptions or planning timely support.

We present StudySense, a context-aware learning agent that treats student-initiated multimodal interactions as sensing events. Camera input, OCR results, questions, corrections, practice records, and review actions are standardized and organized into long-term learning memory. When a new event occurs, the agent retrieves related evidence, infers the current learning state, plans an action, and writes subsequent feedback back to memory.

We instantiate StudySense in a mistake review prototype for middle- and high-school students. The prototype supports problem capture, region selection, OCR, structured analysis, follow-up questioning, targeted exercise generation, and printable review handouts. We describe the formative research, system workflow, implementation logic, evaluation plan, and responsible design boundaries for student controlled learning memory.

*Corresponding author.

UbiComp Companion '26, Shanghai, China

ACM ISBN 979-8-4007-2533-3/2026/10
https://doi.org/10.1145/3798063.3837312

CCS Concepts

• **Human-centered computing** → **HCI design and evaluation methods**; **Ubiquitous and mobile computing systems and tools**; • **Applied computing** → *Computer-assisted instruction.*

Keywords

learning agent; event-based sensing; context-aware computing; long-term memory; generative AI; ubiquitous learning

ACM Reference Format:
Zihao Huang, Tianyi Zhang, Ziyuan Lin, Yuchen Jin, Zihan Cai, Xiaoteng Pan, and Huize Chao. 2026. StudySense: A Context-Aware Learning Agent with Event-Based Sensing and Long-Term Memory. In *Companion of the 2026 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp Companion '26), October 11–15, 2026, Shanghai, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3798063.3837312

1 Introduction

Weiser's vision of technologies that "disappear" into everyday life remains central to ubiquitous computing [4]. In learning, however, generative AI still appears mainly as a tool that students explicitly invoke for a single answer or explanation. Real learning is longitudinal: a mistake made today may reveal its importance only when a similar problem reappears during later practice or exam preparation. One-shot interactions do not preserve enough context to explain why a difficulty recurs or what support should follow.

Context-aware systems adapt to users' tasks, histories, and situations [2, 3], while self-regulated learning emphasizes monitoring, feedback, and adjustment over time [5]. StudySense applies these principles through event based sensing. Instead of continuously monitoring students, it captures deliberate learning interactions from a mobile learning workflow, including camera input, OCR text, questions, corrections, practice requests, and review actions. These heterogeneous events provide evidence about knowledge points, misconceptions, review needs, and transfer failures while keeping sensing visible and student initiated.

StudySense contributes three elements. First, it defines a learning-event representation that connects multimodal input, learning content, intent, time, and relationships. Second, it combines structured, semantic, and relational memory to retrieve evidence and infer a learning state before generating an action. Third, it demonstrates the workflow through a mistake-review prototype grounded in formative research with secondary-school students. The goal is not to evaluate or rank students, but to help them convert fragmented learning activity into reusable, controllable learning memory.

2 Formative Research and Design Requirements

2.1 Method

We conducted a formative survey and semi-structured interviews with students in Grades 7–12 to identify when a learning agent should sense and support everyday learning activity. We distributed 400 questionnaires and received 386 completed responses. All 386 returned questionnaires were retained as valid responses, producing a 96.5% response and valid response rate. The questionnaire covered mistake organization practices, learning pain points, desired support, typical use scenarios, and willingness to continue using the system. Three students representing practice efficiency, knowledge extraction, and paper based review needs then participated in semi-structured interviews. The interviews supported qualitative design interpretation rather than statistical generalization.

2.2 Findings

The survey showed that difficulty continues after an answer is obtained. Time-consuming manual organization was reported by 86% of respondents, uncertainty about how to review by 79%, and lack of targeted practice by 74%. Evening self-study review was the most frequently selected use scenario (92%), followed by exam preparation (85%). These results indicate that students need support for turning mistakes into later practice and review, not only for solving the current problem.

The interviews exposed different forms of the same longitudinal problem. The efficiency-oriented science student described manual mistake organization as repeated labor: paper exercises had to be photographed, classified, copied, and later searched again before similar practice could begin. This student wanted the system to shorten the path from a mistake to exercises that target the same knowledge point and error reason.

The knowledge-extraction-oriented student already used AI for problem search, but reported that explanations sometimes skipped the exact reasoning step that caused confusion. Long videos were inefficient, while exercises with comparable structure were difficult to locate in printed books. This student wanted explanations that could respond to follow-up questions and practice filtered by topic and difficulty. The paper-review-oriented student cared more about accumulation across weeks. Electronic resources were abundant, but few could be reorganized into a stable review path or printable material before an examination.

Across the interviews, three requirements recurred: reduce the effort required to capture paper-based mistakes, connect each event to the underlying knowledge gap, and convert fragmented interactions into reusable review resources. We therefore require low-cost capture from real learning materials, long-term organization of related events, and executable next actions such as clarification, progressive practice, and review-handout generation. These requirements make the agent responsible for continuity across learning events rather than for replacing the student's own judgment.

3 StudySense Agent Design

Figure 1 shows how StudySense transforms student-initiated sensing events into context-aware support and continuously updates learning memory.

3.1 Event-Based Sensing and Standardization

A learning event is a student-initiated interaction that reveals learning context. Events include capturing a problem, asking a follow-up question, correcting an error label, requesting practice, exporting a handout, and completing later review. Each event records its type, timestamp, source, content, subject, knowledge point, intention, evidence, and relationships to earlier events. This schema treats camera and interaction data as contextual signals rather than isolated application commands.

For example, uploading a graph-shift problem creates a problem event containing the image, OCR text, extracted conditions, knowledge point, and an initial error hypothesis. Asking why $x+2$ shifts a graph to the left creates a separate dialogue event linked to the problem. If the student later corrects the error label or requests practice, the correction and request remain distinct events with provenance. The event graph therefore preserves what was observed, what the agent inferred, and what the student confirmed instead of flattening them into a chat transcript.

Raw input is standardized before storage. Images pass through region selection and OCR; extracted text is cleaned and structured into the problem statement, conditions, target question, knowledge point, and possible error reason. Follow-up questions are classified as step-level confusion, conceptual misunderstanding, requests for another explanation, or requests for practice. Privacy filtering removes or marks irrelevant identifiers before an event enters memory.

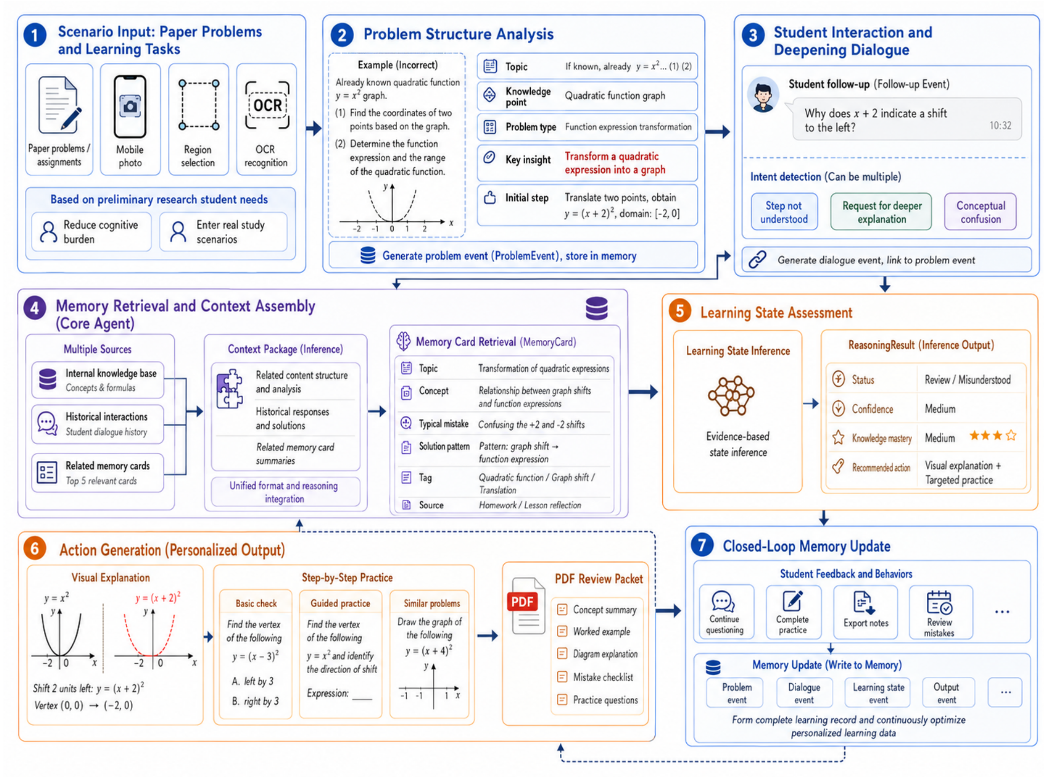

Figure 1: StudySense scenario workflow. Multimodal learning events are standardized, connected to long-term memory, used for evidence-based state inference and action planning, and written back after student feedback.

3.2 Long-Term Memory and Context Retrieval

StudySense uses three complementary memory representations. Structured memory stores explicit fields for filtering by knowledge point, error type, and time. Semantic memory embeds problem summaries, questions, and explanations to retrieve conceptually similar events. Relational memory records links such as shared concepts, repeated error reasons, generated-from relations, and student corrections. Related events are aggregated into memory cards that retain their evidence sources while summarizing recurring concepts and misconceptions.

A memory card centers on a knowledge point, recurring error, or review goal. It contains a compact summary, supporting event identifiers, the student's confirmed corrections, previous actions, and uncertainty. For quadratic-function transformations, a card may connect several problem images, a question about expression-to-graph mapping, an earlier visual explanation, and the outcome of transitional practice. Compression reduces the amount of history passed to the model, while retained identifiers allow the interface to reveal the source of a recommendation.

When a new event occurs, the agent retrieves a bounded context package using structured filters, semantic similarity, and relational expansion. Results are ranked by relevance, recency, user confirmation, and prior use. This approach avoids sending an unfiltered interaction history to the foundation model and allows each suggestion to be traced to concrete learning evidence.

In the prototype, retrieval begins with subject and knowledge point filters, keeps semantic candidates above a 0.72 cosine similarity threshold, expands one relation hop, and returns at most five memory cards plus five source events. Confirmed student corrections are ranked above model hypotheses. The foundation model prompt receives this bounded evidence package and must return JSON containing a state label, evidence identifiers, confidence, and action type.

3.3 State Inference, Action Planning, and Write-Back

The agent uses the retrieved context to distinguish a new problem, repeated error, conceptual misunderstanding, transfer failure, review need, or insufficient evidence. The output contains a candidate state, supporting evidence, confidence, and a recommended action. A planner then selects an explanation, visual clarification, progressive exercise, error summary, or review handout. When evidence is insufficient, the system asks the student to choose the next step rather than presenting an uncertain judgment as fact.

Action planning combines structured rules with foundation model generation. Rules constrain which actions are available for a state and require uncertainty to be surfaced, while the model adapts the explanation, examples, and exercise wording to the retrieved evidence. Typical state-action pairs include new problem to explanation, repeated error to error comparison, conceptual gap to visualization, transfer gap to scaffolded practice, review need to handout generation, and sparse evidence to clarification. Confidence below 0.60 forces the "uncertain" state. Generated content retains its source event identifiers and planned action type, preventing generated suggestions from silently becoming confirmed learner knowledge.

Student feedback becomes new evidence. Follow-up questions are connected to the originating problem and explanation; corrected labels revise earlier hypotheses; generated exercises record their source memory; and completed practice or exported handouts mark later stages of the learning process. This write-back loop allows StudySense to update support over time without relying on continuous background monitoring.

4 Prototype Instantiation

We instantiate StudySense as a mobile mistake-review prototype for middle- and high-school students. A student photographs a paper problem, selects the relevant region, and reviews the OCR result. The system extracts the problem structure, knowledge point, and a possible error reason before generating an initial explanation. The student can ask about a specific step, request similar problems, correct the inferred error label, or export a printable review handout. Each action becomes a linked learning event.

The prototype separates capture, event processing, memory, agent, and output responsibilities. The capture layer produces images, selected regions, and interaction events. Event processing performs OCR cleaning, field extraction, knowledge tagging, intent recognition, and privacy filtering. The memory service maintains structured records, semantic retrieval, and event relations, while the agent service receives only a bounded evidence package. Generated explanations, exercises, and handouts retain links to the events that produced them, allowing later correction and provenance inspection.

The mobile interaction is designed around short, reversible steps. Students first confirm the selected image region and OCR result, then inspect the extracted knowledge point and possible error reason. They can edit either field before requesting an explanation. Follow-up questions remain attached to the relevant reasoning step, and generated exercises are grouped as basic checking, guided transfer, and similar problem practice. Handout export combines the confirmed concept summary, worked example, visual explanation, mistake checklist, and selected practice items into a printable review packet.

Consider a student who asks why $x + 2$ shifts a quadratic graph to the left. StudySense stores the question with the original problem and explanation. When the student later makes a similar error, the agent retrieves the earlier question and practice evidence, identifies a possible repeated conceptual misunderstanding, and recommends a visual explanation followed by transitional exercises. The resulting practice and feedback are written back to the same memory card. This scenario demonstrates how camera sensing, dialogue, memory retrieval, and content generation operate as one longitudinal support process.

5 Evaluation and Responsible Design

StudySense is currently a working prototype, so the planned evaluation targets mechanism validity rather than exam score improvement. A task based study will use multi-event learning scenarios rather than isolated prompts. Participants will enter an initial problem, inspect an explanation, ask a follow-up question, generate practice, correct a label, and later return with a related problem before exporting a handout. The delayed related event is necessary for testing whether memory changes the agent's response.

We will compare the full system with a memory disabled baseline that receives only the current event. Technical measures will include event linking accuracy, retrieval relevance, state classification agreement with expert labels, provenance completeness, latency, and prompt schema validity. Subject experts will rate the correctness, relevance, difficulty progression, misconception tracking quality, and consistency of explanations and exercises. Student facing measures will include task completion, perceived workload, usefulness, confidence in understanding the recommendation, and whether participants can identify the evidence behind it. Interaction logs and interviews will examine when students accept, correct, or ignore an inferred state.

Long term learning memory contains mistakes and inferred weak points, creating risks beyond ordinary chat history. StudySense therefore records only student-initiated events, minimizes captured content, filters incidental identifiers, and keeps memory visible, editable, and deletable. Inferences are presented as suggestions with evidence and confidence, following human-AI interaction guidance on transparency and controllability [1]. Learning memory is restricted to student support and must not be reused for opaque ranking, discipline, or attitude inference. Human confirmation is required before consequential actions or persistent changes to a learner profile.

6 Conclusion

StudySense combines event based sensing, long-term memory, evidence based reasoning, action planning, and feedback in a student controlled learning loop. With clarified retrieval thresholds, prompt contracts, state confidence handling, and evaluation measures, the prototype offers a reproducible path toward context-aware ubiquitous learning support.

References

[1] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–13.

[2] Anind K. Dey. 2001. Understanding and Using Context. *Personal and Ubiquitous Computing* 5, 1 (2001), 4–7.

[3] Bill N. Schilit, Norman Adams, and Roy Want. 1994. Context-Aware Computing Applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*. IEEE Computer Society, Los Alamitos, CA, USA, 85–90.

[4] Mark Weiser. 1991. The Computer for the 21st Century. *Scientific American* 265, 3 (1991), 94–104.

[5] Barry J. Zimmerman. 2002. Becoming a Self-Regulated Learner: An Overview. *Theory Into Practice* 41, 2 (2002), 64–70.