

Always-On Learning Companion: Proactive Multimodal Tutoring for Everyday Study Scenarios

Stanford CS224N {Custom} Project

Haowen Wang[†], Zhen Jia[†], Chenyue Li[†]

[†] Stanford University

haowenw@stanford.edu zhenjia1@stanford.edu chenyuel@stanford.edu

Abstract

Large language models enable AI-assisted learning but remain reactive, requiring users to request help and provide context. We propose a proactive tutoring system that monitors multimodal behavioral signals, such as screenshots and interaction patterns, to infer learner difficulty and provide context-aware guidance. Our architecture combines real-time sensing with a multi-agent tutor that generates concise explanations and visualizations. We evaluate the system in a controlled human study across 70 scenarios, comparing it with a reactive Gemini 2.5 Pro baseline using learning gain, mastery, preference, and interaction time.

1 Key Information to include

- Mentor: Advit Deepak

2 Introduction

AI-assisted learning systems powered by large language models (LLMs) have shown strong potential for providing explanations, answering questions, and supporting problem solving. However, most existing systems operate in a reactive paradigm: users must explicitly formulate questions and provide sufficient context through text prompts or manually upload materials such as screenshots. In real learning scenarios, students often struggle to articulate their confusion or even recognize when they need help, causing many opportunities for timely guidance to be missed.

Proactive support is particularly important in learning environments where user behavior provides rich contextual signals. During problem solving, learners continuously interact with digital environments through actions such as editing documents, navigating materials, or pausing to think. These interactions generate multimodal signals—including screenshots, interaction patterns, and evolving workspaces—that can reveal user intent and potential difficulties. If an AI system could interpret these signals in real time, it could intervene at appropriate moments and provide context-aware guidance. This motivates two key challenges for AI-assisted learning systems: enabling *proactive assistance* and reasoning over *multimodal behavioral information*.

However, addressing these challenges is non-trivial. Behavioral signals are high-frequency, multimodal, and often noisy, making them difficult to interpret reliably. AI agents must also infer latent learner states—such as confusion or misconceptions—from indirect observations rather than explicit queries. Moreover, integrating such signals into the reasoning process of LLMs requires a structured interface that converts raw behavioral data into meaningful contextual representations.

Prior AI tutoring systems primarily focus on text-based conversational agents where users describe their problems through prompts and the AI provides explanations or hints [1]. While some recent systems incorporate visual outputs or interactive elements [2], they still rely on explicit user triggers and manually provided context. As a result, these systems remain fundamentally reactive and do not fully leverage behavioral signals generated during learning.

In this project, we explore proactive human–AI collaboration through an *always-on multimodal tutoring system*. Our system continuously senses user activity through behavioral modules that capture signals such as screenshots and interaction patterns. These signals are transformed into structured observations and provided to a multi-agent tutor architecture that infers learner difficulties and generates concise, context-aware guidance. We evaluate the system through a controlled human study across 70 learning scenarios, comparing it with a reactive baseline based on Gemini 2.5 Pro.

The main contributions of this work are:

- **A proactive AI tutoring framework based on behavioral observation.** We introduce an always-on tutoring system that monitors user activity and proactively intervenes when the learner is likely to be stuck by inferring user intentions and potential difficulties.
- **Multimodal behavioral sensing for AI-assisted learning.** We design a sensing pipeline that processes multimodal signals such as screenshots and interaction patterns to infer user intent and knowledge gaps, enabling context-aware explanations and visualizations.

3 Related Work

Recent work suggests that effective AI-assisted learning requires more than accurate problem solving: it must also provide pedagogically appropriate guidance throughout the interaction. Prior research on tutoring dialogue shows that strong educational support depends on scaffolded explanations, diagnostic questioning, and the ability to guide learners without prematurely collapsing the reasoning process into direct answer revelation [1]. Consistent with this view, interaction-based studies of mathematical assistance argue that the quality of a learning assistant cannot be adequately captured by static input–output benchmarks alone, since educational value emerges through multi-turn collaboration, perceived helpfulness, and the system’s ability to respond constructively to a learner’s evolving needs [3].

Building on these insights, recent multimodal systems further demonstrate that visual and executable explanations can substantially improve conceptual understanding in settings where textual feedback alone is insufficient, highlighting the importance of treating visualization as part of the instructional process rather than as a purely presentational aid [2].

At the same time, emerging work on user modeling from computer-use traces suggests that screenshots and interaction signals can support inference over latent user state and enable proactive assistance without requiring explicit requests from the user [4]. However, these capabilities have largely been explored in isolation: existing work typically focuses either on pedagogically grounded but reactive tutoring, multimodal explanation in bounded task settings, or domain-general proactive assistance rather than support tailored to learning.

Our project brings these strands together in an always-on multimodal learning companion for everyday study scenarios that continuously interprets screen context and behavioral signals, infers likely confusion or hesitation in situ, and proactively delivers timely, pedagogically scaffolded support. Crucially, our system uses visualization not merely as an auxiliary interface feature, but as an adaptive instructional mechanism that can generate or select appropriate visual explanations based on the learner’s current context and likely knowledge gap, thereby supporting understanding when text alone may be insufficient.

4 Approach

Problem Formulation. We study proactive AI-assisted learning, where a system supports users during learning scenarios without requiring explicit help requests. Given a stream of multimodal behavioral signals (e.g., screenshots, interaction patterns, and pauses), the system aims to infer learner difficulties and provide timely, targeted guidance, improving learning efficiency and understanding.

Methods. We propose a agent-based system consisting of a **sensing module** and a **tutor system**.

Sensing Module. The sensing module analyzes on-screen behavior by capturing visual context (e.g., screenshots) and interaction signals, including mouse activity, scrolling, keyboard input, and clicks.

Always-On Learning Companion

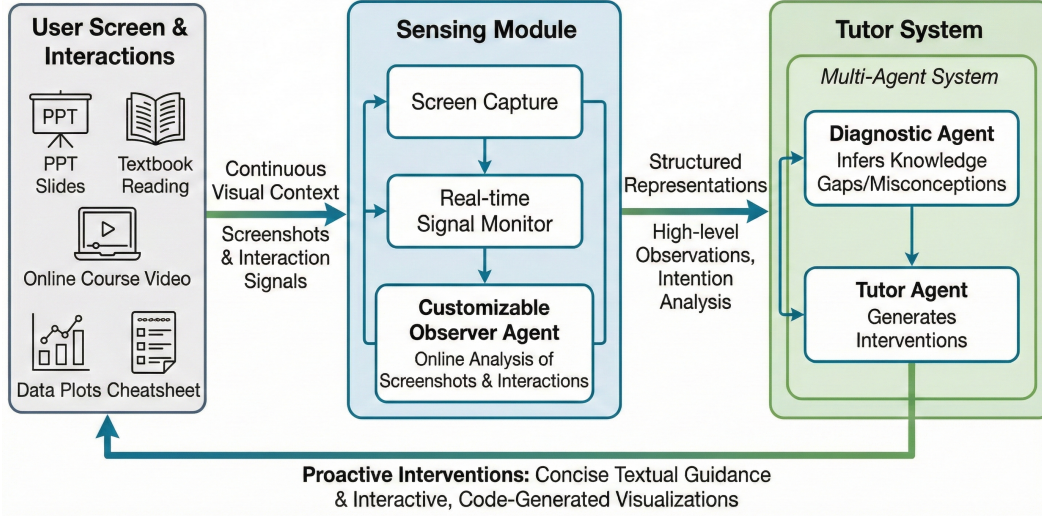


Figure 1: Framework of our always-on learning companion, illustrating the sensing-tutoring pipeline and user interaction loop.

Screenshots are triggered by interaction events rather than continuous recording, preserving privacy and reducing overhead; no webcam-based tracking is used.

This module builds on *Creating General User Models from Computer Use* [4], which supports screen capture and behavioral modeling. While the original GUM framework uses Qwen2.5-VL for general user modeling, our system employs Gemini 2.5 Pro with task-specific prompts for tutoring.

We extend prior work by detecting potential difficulty via inactivity exceeding a configurable idle threshold (default: 2 minutes). To distinguish being *stuck* from *deep reading*, the system combines inactivity with interaction patterns: sustained inactivity without scrolling suggests difficulty, whereas periodic scrolling indicates active reading and suppresses interventions. These signals trigger optional assistance that users may ignore.

Tutor System. The tutor system, implemented entirely by us, is a multi-agent architecture that consumes sensing outputs to provide proactive assistance. A *diagnostic agent* infers potential knowledge gaps or misconceptions, while a *tutor agent* generates concise guidance and interactive, code-generated visualizations. We treat visualizations as first-class interaction primitives that support exploration and conceptual understanding without revealing full solutions.

Baselines. We compare two AI-assisted learning settings that use the same foundation model but differ in interaction paradigm.

Reactive Gemini 2.5 Pro interaction (baseline). Participants interact with Gemini 2.5 Pro via a standard chat interface, requesting help as needed. This reflects the prevalent reactive workflow enabled by modern LLMs.

Proposed proactive teaching agent (our method). Our system also leverages Gemini 2.5 Pro but augments it with sensing and multi-agent tutoring. By continuously monitoring context and proactively initiating guidance, this comparison isolates the impact of observation-driven tutoring from the capabilities of the underlying model.

5 Experiments

5.1 Data

Our evaluation data targets controlled human studies and has two parts: **(I) Topic grouping materials** (7 topics such as calculus, probability, introductory ML, NLP, Computer Vision, data structure, Python) with multiple-choice pre-tests to estimate prior familiarity, balance participants across

System	Avg Pre-test	Avg Post-test	Avg Objective Gain
Our Agent	6.75	8.88	+2.13
Baseline (Gemini)	6.88	7.75	+0.87

Table 1: Average pre-test score, post-test score, and objective gain across all users for the tutoring agent and the baseline system.

System	Avg Familiarity (Before)	Avg Familiarity (After)	Avg Perceived Gain
Our Agent	5.14	8.00	+2.86
Baseline (Gemini)	5.00	5.86	+0.86

Table 2: Average self-reported topic familiarity before and after the learning session, along with the average perceived gain, across all users.

conditions, and compute learning gains via pre/post comparison; and **(II) Evaluation scenarios** curated from public educational resources (slides, textbook excerpts, online course pages/frames, plots, cheatsheets) as *dynamic user action sequences* (e.g., reading slides, watching a course, scrolling a textbook page, interpreting a plot) rather than static screenshots. Users do *not* need to upload screenshots/videos: a sensing module continuously captures on-screen context and interaction signals, logs time-stamped actions with screenshots to a database, and every 20 seconds pulls new actions and segments them into coherent “context windows” using screenshot similarity (MSE distance) for downstream tutoring.

5.2 Evaluation Method

We conduct a controlled user study comparing **free-form LLM chat** (reactive baseline using Gemini 2.5 Pro) and **our always-on tutoring agent** (proactive monitoring, confusion detection, concise guidance, and code-generated visualizations).

All participants first complete a pre-test to assess their prior knowledge of the topics. Based on the results, each participant is assigned two topics that they are not familiar with. One topic is studied using our tutoring agent and the other using the baseline system. To reduce topic-related bias, topics are assigned in a balanced manner across participants so that both systems are evaluated on a comparable set of topics.

We evaluate learning outcomes using several complementary metrics. First, we measure learning improvement using **objective gain**, defined as:

$$\text{Objective Gain} = S_{\text{post}} - S_{\text{pre}} \quad (1)$$

where S_{pre} and S_{post} denote the pre-test and post-test scores.

Second, we analyze the **distribution** of post-test scores and objective gains across participants. Third, we measure **perceived gain**. Before and after each learning session, participants report their topic familiarity on a 1–10 scale to capture perceived mastery. We define perceived gain as:

$$\text{Perceived Gain} = F_{\text{post}} - F_{\text{pre}} \quad (2)$$

where F_{pre} and F_{post} denote the self-reported topic familiarity before and after the learning session.

In addition, we collect **user feedback** through a post-study questionnaire to assess participants’ perceptions of the learning experience. The survey includes questions on the perceived helpfulness and clarity of the system, the usefulness and reliability of generated visualizations, and whether participants would consider using the tool in future learning scenarios.

Together, objective and subjective measurements allow us to evaluate both learning effectiveness and user experience when comparing the baseline paradigm with our tutoring-oriented agent.

User Feedback Metric	Agent	Gemini
Helpful	4.43	2.71
Clear explanations	4.29	2.86
Visualization helpful	4.43	–
Would use again	4.71	2.57

Table 3: User feedback scores (1–5 Likert scale). Participants rate the tutoring agent substantially higher than the Gemini baseline across all comparable dimensions.

5.3 Experimental Details

Sensing frequency and segmentation. All actions are logged with screenshots; every 20 seconds, the system pulls actions and segments them using MSE-based visual distance between consecutive screenshots, producing stable context windows for diagnostic/tutor agents and supporting both (i) online proactive nudges and (ii) offline analysis of intervention timing/usefulness. **Model and deployment details.** Both conditions are implemented by us via the model APIs; our agent uses a strong multimodal LLM backbone (Gemini 2.5 Pro) to tutor and reason over sensed context, we reuse no published scores, and all baseline results come from our own matched-condition user study.

5.4 Results

Tables 1 and 2 summarize the aggregate results for objective performance and self-reported familiarity. Following Equation 1, our agent achieves a substantially larger average objective gain than the Gemini baseline. The average pre-test score increases from 6.75 to 8.88, corresponding to an average objective gain of **+2.13**, whereas Gemini improves from 6.88 to 7.75, corresponding to an average objective gain of **+0.87**. A similar pattern is observed for perceived learning outcomes. Following Equation 2, the average familiarity rating under our agent increases from 5.14 to 8.00, giving an average perceived gain of **+2.86**, while Gemini increases from 5.00 to 5.86, corresponding to a perceived gain of **+0.86**.

We also collected subjective user feedback after each learning session (Table 3). Participants consistently rate the tutoring agent higher than the Gemini baseline in terms of helpfulness, clarity of explanations, and willingness to use the system again. In addition, the visualization component of our system receives high ratings from users, suggesting that visual explanations contribute positively to the learning experience. Visualization feedback is only reported for our system, since the Gemini baseline interaction does not provide visualization-based explanations.

Overall, these results are broadly consistent with our hypothesis and somewhat stronger than we initially expected, especially for perceived gain. We hypothesize that this advantage comes from two aspects of our approach. First, the system proactively intervenes when it detects likely confusion from screen context and interaction signals, rather than waiting for the learner to formulate a question. Second, the system can provide targeted hints and appropriate visualizations tailored to the learner’s likely knowledge gap, making the assistance more interpretable and pedagogically effective than a reactive text-only baseline. Taken together, these findings provide preliminary evidence that proactive, context-aware, and visualization-supported tutoring can improve both objective learning outcomes and learners’ perceived mastery relative to a strong baseline.

6 Analysis

Beyond quantitative metrics, we conduct qualitative analysis to better understand the behavior of our tutoring agent. We analyze examples, evaluate performance across different data subsets, and compare the behavioral characteristics of our system with a baseline system.

6.1 Analysis of Objective Gain

Figure 2 summarizes the average objective and perceived gains across conditions. Objective gain and perceived gain are computed according to Equations 1 and 2. Error bars denote ± 1 standard error of the mean (SEM). We report the mean gain across samples, with error bars indicating ± 1

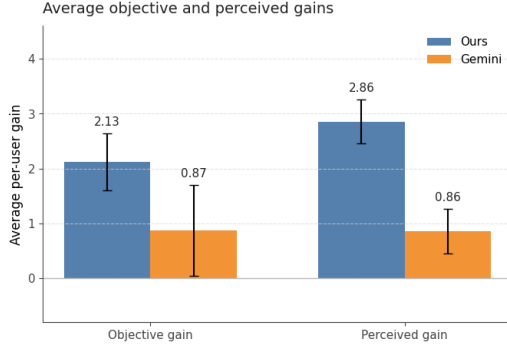


Figure 2: Average objective and perceived gains per user.

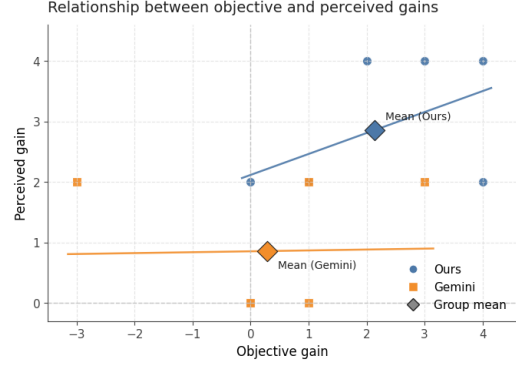


Figure 3: Objective gain versus perceived gain across paired samples.

standard error of the mean (SEM). Our tutoring agent achieves substantially higher gains than the Gemini 2.5 Pro baseline on both objective and perceived measures.

Figure 3 further examines the relationship between objective and perceived improvements. Each point represents one paired sample, with objective gain on the x-axis and perceived gain on the y-axis. Samples from our system tend to appear in the upper-right region of the plot, suggesting that larger improvements in objective performance are often accompanied by larger increases in perceived understanding. In contrast, Gemini samples are more concentrated in lower-gain regions, indicating weaker joint improvements across the two dimensions. We hypothesize that this may be due to our agent’s extensive use of visualizations, which support more intuitive understanding and increase participants’ confidence in their topic familiarity.

Metric	Agent	Gemini
Perfect Score (10/10)	50.0%	12.5%
Gain ≥ 3 points	37.5%	25.0%
Gain ≤ 0	12.5%	50.0%

Table 4: Distribution of objective gain.

Beyond average scores, we also analyze the distribution of objective gain (Table 4). Students interacting with the tutoring agent are more likely to achieve high post-test scores. In particular, perfect scores occur more frequently in the agent condition. Furthermore, large learning gains (three points or more) appear more often when students learn with the agent.

We also observe that students with lower initial familiarity (pre-test scores around 5–6) tend to benefit more from the tutoring agent, suggesting that guided explanations and interactive tutoring strategies may provide the greatest benefit for learners with moderate prior knowledge.

6.2 Selected Case Studies

To better illustrate how the tutoring agent supports student learning, we present several representative interaction examples. Each example highlights a specific pedagogical capability of the system.

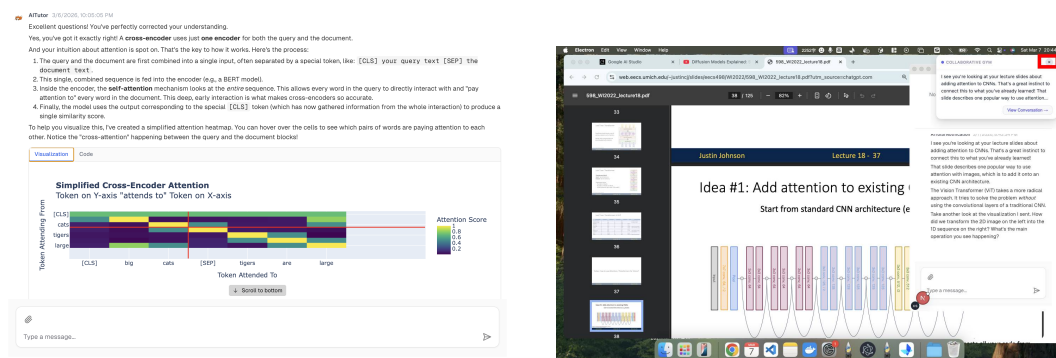
In particular, these examples demonstrate three key behaviors of the tutoring agent: (1) **visualization-assisted explanations**, where the agent proactively generates visual diagrams to clarify abstract concepts; (2) **proactive confusion detection**, where the agent identifies potential misunderstandings during the learning process.

Example 1: Visualization-Assisted Explanation This example shows how the tutoring agent uses visualizations to support conceptual understanding during interactive learning.

The student first asks: “What is a cross-encoder?” The agent provides a conceptual explanation and generates a visualization of the retrieve–re-rank pipeline used in modern retrieval systems (Figure 4a).



(a) Visualization of the retrieve-re-rank pipeline illustrating the role of the cross-encoder. (b) Comparison between bi-encoder and cross-encoder architectures.



(c) Attention heatmap illustrating token interactions in a cross-encoder. (d) Example of proactive tutoring assistance. The agent provides a contextual hint to offer guidance.

Figure 4: Case studies from our tutoring system. Top row (a–b) and bottom-left (c) show visualizations generated during interaction. Bottom-right (d) shows an example of proactive confusion detection.

The student then asks: “Does a cross-encoder use two encoders, one for the query and one for the document?” The agent responds with a diagram comparing the student’s imagined architecture with the actual cross-encoder architecture, highlighting how they differ in processing queries and documents (Figure 4b). Compared to directly presenting the cross-encoder architecture, this visualization allows the student to intuitively identify where their understanding diverges from the mechanism.

Finally, the student asks: “So for a cross-encoder, it only has one encoder? And the relevance score is computed based on attention?” The agent generates an attention heatmap to illustrate how token-level interactions between the query and document contribute to the relevance score (Figure 4c).

Across the interaction, visualizations are generated together with textual explanations to make abstract concepts easier to understand.

Example 2: Proactive Confusion Detection The tutoring agent also provides proactive assistance by monitoring the student’s interaction context during learning. When the system detects that the student has remained on the same material for an extended period without further interaction, it treats this as a possible sign of difficulty.

Figure 4d shows a student reading lecture slides on attention mechanisms in convolutional neural networks. After observing that the student stays on the same slide for a prolonged period, the agent sends a contextual hint through a notification panel. The message acknowledges the current topic and offers a short explanation that connects it to previously learned material.

Rather than waiting for an explicit question, the agent provides timely guidance and invites further interaction. This helps reduce situations in which students remain stuck without asking for help, creating a more responsive and interactive learning experience.

6.3 Behavioral Comparison with Baseline Systems

Beyond performance metrics, we compare the interaction behaviors of our tutoring agent with a baseline conversational system, as illustrated in Table 5.

Aspect	System Behavior
Answer Style	Baseline: direct answer Agent: step-by-step guidance
Confusion Handling	Baseline: reactive explanation Agent: proactive clarification
Visualization	Baseline: text-only responses Agent: generates visual explanations
Interaction Strategy	Baseline: single-turn responses Agent: interactive tutoring dialogue

Table 5: Behavioral comparison between the baseline system and our tutoring agent.

Compared to the baseline system, which primarily provides direct answers, our tutoring agent adopts several pedagogical interaction strategies. The agent provides step-by-step guidance instead of immediately revealing the final answer, encouraging students to reason through the problem. It also detects potential confusion and asks clarification questions before giving explanations.

In addition, the agent can generate visualizations to complement textual explanations and illustrate abstract concepts. Finally, the system supports multi-turn tutoring dialogue, guiding students through reasoning steps rather than responding with a single answer.

Overall, these behaviors emphasize guided reasoning, visualization, and proactive clarification, which more closely resemble human tutoring interactions.

7 Conclusion

We present an always-on tutoring agent designed to support interactive learning through proactive monitoring, confusion detection, step-by-step guidance, and visualization-assisted explanations. Unlike traditional reactive LLM chat interfaces, our system adopts a tutoring-oriented interaction paradigm that emphasizes guided reasoning and interactive dialogue.

Results from a controlled user study show that participants using our agent achieve larger learning gains and higher post-test scores compared to a baseline free-form LLM interaction. Qualitative analysis further indicates that visualization generation, proactive clarification, and structured guidance help students better understand abstract concepts.

These findings suggest that tutoring-oriented interaction strategies can improve the effectiveness of LLM-assisted learning systems. Future work will explore larger-scale evaluations and more advanced mechanisms for adaptive tutoring and personalized guidance.

Team Contributions

1. Haowen Wang: Implemented the method and carried out the evaluation experiments.
2. Zhen Jia: Designed and carried out the evaluation experiments, and performed the analysis.
3. Chenyue Li: Designed and carried out the evaluation experiments, and performed the analysis.

References

- [1] Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. Mathdial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, 2023.

- [2] Jimin Lee, Steven-Shine Chen, and Paul Pu Liang. Interactive sketchpad: A multimodal tutoring system for collaborative, visual problem-solving. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2025.
- [3] Katherine M Collins, Albert Q Jiang, Simon Frieder, Lionel Wong, Miri Zilka, Umang Bhatt, Thomas Lukasiewicz, Yuhuai Wu, Joshua B Tenenbaum, William Hart, et al. Evaluating language models for mathematics through interactions. *Proceedings of the National Academy of Sciences*, 121(24):e2318124121, 2024.
- [4] Omar Shaikh, Shardul Sapkota, Shan Rizvi, Eric Horvitz, Joon Sung Park, Diyi Yang, and Michael S Bernstein. Creating general user models from computer use. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, pages 1–23, 2025.

A Appendix

This appendix provides examples of the evaluation materials used in our user study, including the pre-test forms for different domains and the post-study feedback form. Figures 5, 6, and 7 show representative screenshots of the pre-test forms used to assess participants’ objective performance before the study session. Figure 8 shows the feedback form used to collect participants’ post-study subjective evaluations, including perceived familiarity, preference, and qualitative comments. For completeness and reproducibility, we also include links to all released forms below.

Pre-test form links.

- **NLP:** Google Form link
- **CV:** Google Form link
- **Data:** Google Form link
- **Python:** Google Form link
- **Machine Learning:** Google Form link
- **Probability:** Google Form link
- **Calculus:** Google Form link

Post-study feedback form link.

- **User feedback:** Google Form link

1. What is the primary purpose of the BPE (Byte Pair Encoding) tokenization algorithm widely used in modern Large Language Models (like GPT and LLaMA)? *
- ☐ A. To construct a subword vocabulary balancing fixed dictionary sizes with open-vocabulary coverage.
 - ☐ B. To map discrete semantic tokens into continuous, fixed-length vector representations for embedding.
 - ☐ C. To dynamically compress sequence length by filtering out high-frequency stop words and punctuation.
 - ☐ D. To assign unique integer identifiers to distinct semantic concepts via hierarchical clustering.
-
2. In the Transformer architecture, what is the relationship between the time/space complexity of the Self-Attention mechanism and the input sequence length N ? *
- ☐ A. $O(N)$
 - ☐ B. $O(N \log N)$
 - ☐ C. $O(N^2)$
 - ☐ D. $O(2 \cdot N)$
-
3. Compared to traditional absolute position encoding, what is the main reason for the massive success of RoPE (Rotary Position Embedding) in modern LLMs like LLaMA? *
- ☐ A. It modifies self-attention weights directly by adding a learnable scalar bias to the attention matrix.
 - ☐ B. It injects absolute sinusoidal signals into the token embeddings before the first transformer block.
 - ☐ C. It implicitly achieves relative position encoding by applying a rotation matrix to queries and keys.
 - ☐ D. It computes relative distances between tokens and feeds them through an independent neural network.

Figure 5: Example NLP pre-test form used to evaluate participants' objective performance before the learning session.

6. What is the mathematical essence of the Forward Process in Diffusion Models? *

- ☐ A. An autoencoding process that maps a raw image into a regularized continuous latent space.
- ☐ B. A predefined Markov Chain that gradually adds Gaussian noise to the target data distribution.
- ☐ C. A deterministic iterative process of gradually removing predicted noise from a corrupted image.
- ☐ D. An adversarial min-max game played sequentially between a generation network and a discriminator.

7. The CLIP (Contrastive Language-Image Pretraining) model is pre-trained on massive image-text pairs. What is its core loss function? *

- ☐ A. A Triplet Margin Loss anchoring on the image and using text descriptions as positive or negative samples.
- ☐ B. A Masked Language Modeling loss combined with a masked image patch reconstruction loss.
- ☐ C. A pixel-level Mean Squared Error loss applied directly to the reconstructed input images.
- ☐ D. A Contrastive Loss maximizing cosine similarity for matching pairs and minimizing it for others.

8. NeRF (Neural Radiance Fields) uses a Multi-Layer Perceptron (MLP) to implicitly represent 3D scenes. What are the inputs to this MLP? *

- ☐ A. Multi-view depth maps and their corresponding surface normal vectors.
- ☐ B. 3D voxel grids and their corresponding volumetric density values.
- ☐ C. The 3D spatial coordinates (x, y, z) and the viewing direction (θ, ϕ) .
- ☐ D. 2D image pixel coordinates and the camera's intrinsic focal lengths.

Figure 6: Example CV pre-test form used in the same evaluation pipeline.

5. What is the main difference between Covariance and Correlation? *

- ☐ Covariance is always positive, while correlation can be negative.
- ☐ Covariance depends on data scale, while correlation is normalized.
- ☐ Correlation applies to discrete data, while covariance applies to continuous data.
- ☐ Covariance and correlation are mathematically unrelated.

6. What type of event is the Poisson Distribution most commonly used to model? *

- ☐ The number of heads obtained in a small number of coin flips.
- ☐ The number of rare events occurring in a fixed time interval.
- ☐ Random measurement errors in experimental observations.
- ☐ Long-term trends observed in stock market prices.

7. The Central Limit Theorem (CLT) states that, given a sufficiently large sample size, the ___ of a large number of independent and identically distributed random variables will approach a Normal Distribution, regardless of the variables' original distribution? *

- ☐ Product
- ☐ Maximum value
- ☐ Sample mean
- ☐ Variance

Figure 7: Example Probability pre-test form used to measure objective learning outcomes across domains.

Test Score

Which topic did you study in this experiment? *

☐ Natural Language Model

☐ Computer Vision

☐ Data Structure

☐ Python

☐ Machine Learning

☐ Probability

☐ Calculus

What was your pre-test score before using the tool? *

12345678910

☆

☆

☆

☆

☆

☆

☆

☆

☆

☆

What was your post-test score after using the tool? *

12345678910

☆

☆

☆

☆

☆

☆

☆

☆

☆

☆

Figure 8: Post-study feedback form used to collect subjective ratings and qualitative user feedback after interaction with the system.

12