Contexty: Capturing and Organizing In-situ Thoughts for Context-Aware AI Support

Yoonsu Kim
School of Computing
KAIST
Daejeon, Republic of Korea
yoonsu16@kaist.ac.kr

Chanbin Park
EECS
University of California, Berkeley
Berkeley, California, USA
chanbin.park@berkeley.edu

Kihoon Son
School of Computing
KAIST
Daejeon, Republic of Korea
kihoon.son@kaist.ac.kr

Saelyne Yang
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA
saelyney@andrew.cmu.edu

Juho Kim
School of Computing
KAIST
Daejeon, Republic of Korea
SkillBench, Inc.
Santa Barbara, California, USA
juhokim@kaist.ac.kr

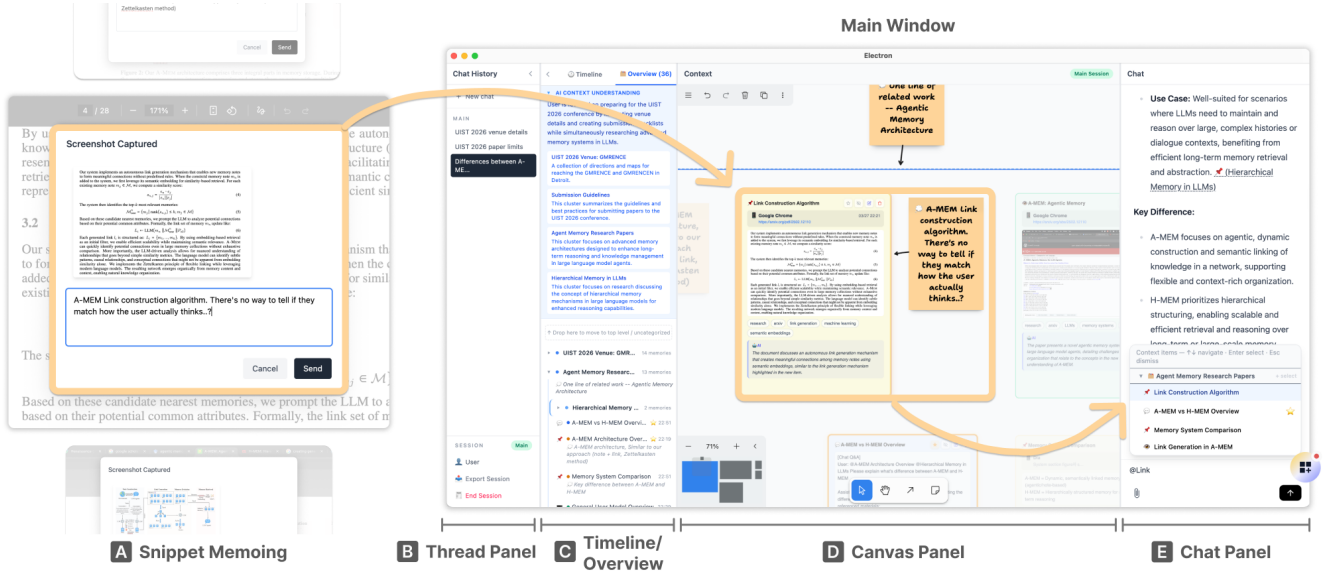

Figure 1: Overview of Contexty. Users capture their in-situ thoughts during their task via Snippet Memoing (A), which are fed into the AI's context. This context is visualized on the Canvas Panel (D) in an inspectable, correctable form, allowing users to review, reorganize, and refine how the AI has represented their task context. Users can also navigate this context through a Timeline or hierarchical Overview (C), and interact with an AI agent grounded in this context via the Chat Panel (E).

Abstract

During complex knowledge work, people engage in iterative sense-making: interpreting information, connecting ideas, and refining their understanding. Yet in current human–AI collaboration, these cognitive processes are difficult to share and organize for AI. They arise in situ and are rarely captured without interrupting the task, and even when expressed, remain scattered or reduced to system-generated summaries that fail to reflect users' cognitive processes. We address this challenge by enabling AI context that is grounded in users' cognitive traces and can be directly inspected and revised by the user. We first explore this through a probe system that supports in-situ snippet memoing, allowing users to easily share their cognitive moves. Our study (N=10) highlights the value of capturing such context and the challenge of organizing it once accumulated. We then present Contexty, which supports users in inspecting and refining these contexts to better reflect their understanding of the task. Our evaluation (N=12) showed that Contexty improved

UIST '26, Detroit, MI, USA

ACM ISBN 979-8-4007-2856-3/26/11
https://doi.org/10.1145/3830398.3830663

task awareness, thought structuring, and users' sense of authorship and control, with participants preferring snippet-grounded AI responses over non-grounded ones (78.1%). We discuss how capturing and organizing users' cognitive context enables AI as a context-aware collaborator while preserving user agency.

CCS Concepts

• **Human-centered computing → Interactive systems and tools**.

Keywords

Human-AI Collaboration, Context Sharing, User-Inspectable AI Context, Context-Aware AI Agents

ACM Reference Format:

Yoonsu Kim, Chanbin Park, Kihoon Son, Saelyne Yang, and Juho Kim. 2026. Contexty: Capturing and Organizing In-situ Thoughts for Context-Aware AI Support. In *The 39th Annual ACM Symposium on User Interface Software and Technology (UIST '26), November 02–05, 2026, Detroit, MI, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3830398.3830663

1 Introduction

Much of the reasoning that shapes complex tasks emerges in fragmented, moment-to-moment cognitive moves [31]. As users read a paper, compare alternatives, or piece together information across sources, they continuously form and revise judgments—why a finding feels significant, how options are being weighed, or what criteria have just shifted. However, these fleeting thoughts are difficult to share and organize into current AI systems. They arise during the task, but articulating them requires interrupting the task to write a prompt. Even when expressed, they remain scattered across chat turns rather than organized into a structured, user-accessible form. As a result, users struggle to reconstruct what has been considered or why, while the AI continues to operate on context that is misaligned with the user's current understanding of the task, offering suggestions that are plausible but no longer useful.

While recent work shows that passive observation of users' on-screen activity can provide rich contextual signals [32, 44], behavioral traces reveal what a user did and where they spent time, but cannot capture the user's internal criteria or interpretations behind those actions—the *why* and *how* of what they did [49]. Similarly, recent LLM systems' automatic memory or context summarization [2, 29] distill interaction history into user-readable summaries. However, these summaries reflect system-generated abstractions rather than how users prioritize information or organize relationships between ideas, making it difficult for users to inspect or understand how their task is currently represented.

We frame this as two interrelated challenges. First, a **capturing problem**: users lack lightweight ways to externalize their in-situ reasoning as they work, so fleeting judgments, comparisons, and interpretations disappear rather than becoming part of the shared context. Second, a **visibility problem**: even when some context is shared, it remains buried in chat history or reduced to system-generated summaries, leaving neither side with a clear picture of how the user's thinking about the task has progressed. These challenges echo long-standing insights from HCI and cognitive science: complex work is shaped by in-the-moment reasoning that disappears if not externalized [31], and understanding only deepens when captured thoughts can be organized and revisited [11, 27, 52].

Building on these insights, we propose a different approach to context sharing in human–AI collaboration, giving users lightweight affordances to externalize their reasoning in-situ and to inspect and refine the AI context. We first explored this idea through a probe system that allowed users to capture screen snippets and attach short in-situ memos directly into an LLM's context. We deployed the probe with 10 experienced LLM users on real-world sensemaking tasks and found that in-situ snippet memoing enabled richer, lighter, and more forward-looking context sharing than prompting. However, the study also revealed that capturing alone was insufficient. As snippets accumulated, participants wanted to organize these evolving cognitive traces—grouping related ideas, tracking changing criteria, and connecting information across sources—while also understanding how this evolving context was represented and used by the AI.

To address these challenges, we present Contexty, a system that externalizes AI context as a user-inspectable and correctable representation grounded in users' cognitive processes. Building on the probe, Contexty introduces a canvas-based context space that organizes AI context constructed from users' in-situ snippet memos and passive computer-use observations. The canvas serves both as an inspectable view of the context AI uses to ground its responses and as the user's sensemaking workspace. Because this context is continuously assembled from users' ongoing task activity and in-situ thoughts, inspecting and refining AI context becomes a natural part of doing the task.

We evaluated Contexty in a within-subjects study with 12 participants on real-world exploratory tasks. Our results show that Contexty significantly improves users' task awareness, thought structuring, perceived AI understanding, and their sense of authorship and control compared to a baseline condition. Participants also consistently preferred AI responses generated with snippet context over those without (78.1%), demonstrating that users' in-situ thoughts provide meaningful grounding for more relevant AI support. These findings point to a broader opportunity: when AI context is grounded in user-authored cognitive traces and made inspectable and correctable, human–AI collaboration can become more transparent and more aligned with users' ongoing cognitive processes during the task. At the same time, our study surfaces important trade-offs between the effort of externalization, perceived usefulness, and users' sense of agency. Based on these findings, we discuss how future systems can capture user context with minimal burden while supporting its organization, inspection, and reuse over long-term human–AI collaboration.

2 Related Work

2.1 Context Sharing for Human-AI Collaboration

Effective human-AI collaboration relies on shared context between users and AI systems. To provide relevant and aligned support, AI systems must understand not only users' explicit requests but also the broader context and underlying intent that shape those requests. Prior work has explored various approaches to how user context can be constructed and utilized in human-AI collaboration.

One line of work focuses on interaction techniques to better elicit and align user intent, including scaffolding prompt formulation [3], supporting iterative refinement [45], and representing intent as structured or manipulable units and as evolving, revisable configurations across interaction [8, 16]. These approaches improve how users articulate and maintain their intents, but still rely primarily on explicitly expressed inputs. Another line of work focuses on how AI context is constructed, maintained, and utilized. In context engineering, researchers have explored incorporating broader context, such as retrieved documents, external knowledge, and interaction history, into model inputs to improve grounding and response quality [26]. In parallel, recent work has enabled users to supply and curate contextual knowledge that can be dynamically integrated during interaction [56], or embedding AI into shared workspaces to make interactions more visible and situated [21]. Beyond supplying context, some work makes the AI's grounding context itself explicit and user-manageable by reifying past conversations into memory objects that users can retrieve and reuse [51], or letting users commit new intent into a specification that grounds the AI's responses, with support for resolving conflicts that arise [43]. Agent-based approaches further incorporate continuous interaction traces and environmental signals, enabling AI systems to proactively gather and utilize context beyond explicit user input [23, 48]. Our work builds on this direction but focuses on a different form of context. Rather than retrieved information, prior conversations, user-authored specifications, or behavior traces, we investigate how AI context can be continuously assembled from users' ongoing task activity and cognitive processes. We further explore how this context can be represented in a form that users can inspect, organize, and refine as they work, supporting both users' evolving understanding of the task and the AI's grounding.

2.2 Externalizing In-situ Cognitive Moves in Knowledge Work

Externalizing thoughts plays a central role in knowledge work, enabling people to offload cognitive effort and refine their understanding over time [11, 27]. Prior work further suggests that much of this reasoning unfolds as in-the-moment activity (e.g., reflection-in-action), where users continuously interpret, compare, and adjust their understanding during a task [31].

Prior systems have explored supporting thought externalization through different interaction designs. Early approaches drew on think-aloud methods that externalize ongoing thought processes [17]. Building on this idea, recent systems support in situ captures integrated with ongoing activity. For example, *PointAloud* [9] enables users to articulate reasoning alongside pointer-based interactions, while *ClearFairy* [36] captures intermediate reasoning through in-situ questioning and rationale inference. *LiquidText* [41] and related annotation-based work [24] support linking ideas within reading workflows; and *Fuse* [18] enables collecting and comparing information during browser-based exploration. More recently, *Orality* [22] further reduces the effort of capturing early-stage thoughts by transforming them into structured representations. Despite these advances, prior approaches do not support using in-situ cognitive moves as a shared resource for working with AI, limiting their role in maintaining aligned task context over time; we address this by enabling their capture, organization, and collaborative use in human–AI interaction.

2.3 Organizing Externalized Thoughts in Knowledge Work

Externalizing thoughts alone is insufficient unless they can be organized and revisited over time. Schön's notion of *reflection-on-action* emphasizes the importance of returning to externalized artifacts to interpret, restructure, and deepen understanding [31]. Similarly, prior work on external representations shows that organizing information into structured forms reduces cognitive load and makes implicit relationships more explicit [1, 52].

Prior work has explored supporting such organizations in various ways. Systems such as *ConceptScope* and *ConceptEVA* support iterative refinement of users' conceptual understanding, helping users evolve their interpretation of a topic over time [53, 54]. In the context of LLM interaction, systems such as *Sensecape* and *Graphologue* structure outputs into visual or relational representations for comparison [13, 38], while *Luminate* and *Policy Maps* map outputs into structured spaces to support systematic analysis [19, 37]. Beyond LLM outputs, *Orca* treats webpages as malleable materials that users and AI can collaboratively organize into a dynamic browser-level workspace [14]. In scholarly sensemaking, *ScholarMate* and *Synergi* integrate AI into visual workspaces for organizing literature and qualitative codes to support collaborative analysis and synthesis [15, 50]. Recent work also explores shared representations for organizing evolving context in collaboration [55]. While these systems support organizing information, they typically focus on externally available content such as model outputs, documents, or literature. Our work instead focuses on organizing users' own evolving task context by incorporating users' in-situ thoughts and ongoing task activity into a shared context, grounding AI support in users' cognitive processes.

3 Design Exploration: Understanding Cognitive Externalization via Snippet Memoing

We begin by exploring how users share their in-situ thoughts with an AI as they work. Much of this reasoning—interpreting information, comparing options, and forming tentative judgments—emerges in the moment, but these fleeting cognitive moves are rarely captured or shared with the AI. In particular, prompting requires users to reformulate these thoughts as explicit instructions, which introduces friction and often filters out process-level reasoning. To explore this, we introduce ***snippet memoing***, a lightweight interaction method that allows users to capture and share fragments of information on screen along with their associated thoughts in situ, directly into the LLM's context.

3.1 Research Probe

We developed a desktop-based LLM application as a probe that enables users to capture and share their in-situ thoughts while working on a task. The application registers global keyboard shortcuts (`Cmd/Ctrl+Shift+S` for screenshots, `Cmd/Ctrl+Shift+C` for text) that can be triggered from any application, without switching applications or interrupting users' workflow. A pop-up immediately appears showing the captured content alongside a text field where

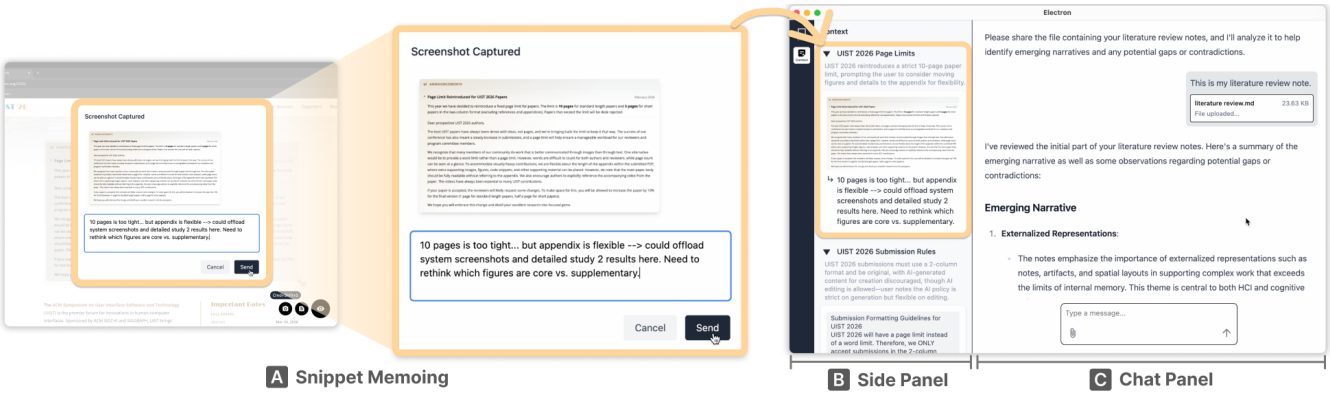

Figure 2: The probe system for snippet memoing. (A) When a user captures a screenshot snippet, a pop-up appears showing the captured content alongside a text field for attaching an in-situ memo. (B) Sent snippet–memo pairs accumulate in a side panel, where each entry displays the captured content and the user's memo, alongside an AI-generated title and summary reflecting how the system interpreted the snippet. (C) The chat panel on the right allows users to interact with the LLM, which generates responses grounded in the accumulated context.

users can attach a short memo: a brief thought, interpretation, or rationale tied to what is on screen at that moment (Figure 2A).

Once sent, these snippet–memo pairs accumulate in a persistent side panel, where each entry displays the captured content and the user's memo, alongside an AI-generated title and summary reflecting how the system interpreted the snippet (Figure 2B). All accumulated snippets are included in the LLM's context window, and users can interact with the LLM through a chat panel to ask questions or request assistance grounded in this context (Figure 2C). The system was built with Electron [7] and TypeScript, compatible with macOS and Windows, and used `gpt-4o-2024-08-06` as the underlying LLM. Participants installed the system on their personal laptops to preserve their natural work environment.

3.2 Participants and Procedure

We recruited 10 participants (5 male, 5 female; age $M = 26$, age $SD = 1.87$) with extensive daily LLM experience (e.g., ChatGPT, Claude, Gemini), with four participants using them every day and six using them 2-5 times a week. Participants were recruited through online recruitment postings on our university community platform. The task was designed to elicit exploratory, multi-source sensemaking, where users interpret information, compare options, and iteratively refine their understanding, allowing in-situ thought capture to naturally arise. Participants explored potential labs and advisors to build a shortlist, and then drafted a cold email to one selected professor. To ensure ecological validity, we pre-screened participants who were actively considering applying to MS, Ph.D., or postdoctoral programs, confirming that the task would be personally relevant and motivating. Each session began with a 10-minute system tutorial, followed by a 30-minute task session in which participants used the system to complete the task while we observed their process, and concluded with a 20-minute post-survey and semi-structured interview. The survey included NASA-TLX and 7-point Likert-scale items on overall satisfaction, perceived context utilization, and burden of sharing the context. The interview explored how this form of interaction differed from conventional prompting, what types of thoughts participants captured and why, and what design opportunities or pain points they perceived. All sessions were screen-recorded and transcribed for later analysis. Participants were compensated KRW 20,000 (≈ USD 14) for their time, and the study was approved by our institutional IRB.

3.3 Findings: How Snippet Memoing Supports Cognitive Externalization

Our findings show that snippet memoing enables different context sharing compared to traditional prompting, capturing not only what users explicitly ask, but also their ongoing interpretations, partial conclusions, and task-relevant reasoning as they unfold.

3.3.1 ***Expressing user-framed perspectives and forward-looking thoughts.*** Snippet memoing enabled participants to externalize not just *what* they were looking at, but *how* they were making sense of it in the moment. By attaching a memo to each captured snippet, users expressed their own interpretations, emerging criteria, and tentative judgments—what felt significant, what remained uncertain, or how it might be used. This created a form of user-framed context, where each snippet reflected the user's cognitive process at that moment. P2, P5, and P8 emphasized that this allowed the system to better align its responses with their intentions. This benefit was also reflected in the post-survey: participants rated the item "I felt that the system made good use of the information I provided with a better understanding of my context" with an average of 6.0/7 (SD = 0.67). Participants also used snippets to capture forward-looking thoughts as they arose. Rather than waiting to formulate a concrete question, they externalized observations and partial ideas that they anticipated to be relevant later, even when they were not yet sure how (P4–6, P8). This distinction between "what I want to remember or use later" and "what I want to ask" gave participants greater flexibility in their interactions, in contrast to the prompt-only method, where context must be compressed into a single turn.

3.3.2 ***Lower barrier to externalizing thoughts in situ.*** Participants (P2–3, P5–6, P8–10) emphasized that snippet memoing lowered the cognitive and interactional cost of sharing context. Rather than pausing to compose elaborate prompts or switch applications, they could capture and annotate their thoughts continuously while working. P6 and P8 noted that because they could explicitly choose where and how to share their thoughts, there was less psychological burden compared to composing a full prompt. This ease was also reflected in the post-survey: when asked *"The way I shared what I was looking at or thinking about to the system didn't feel burdensome"*, participants rated it with an average of 6.6/7 ($SD = 0.66$). NASA-TLX scores further confirmed that the interaction did not impose substantial workload: mental demand ($M = 2.5$), effort ($M = 2.4$), and frustration ($M = 1.3$) all remained low, while perceived performance was high ($M = 5.6$).

3.3.3 ***Captured snippets as cognitive scaffolds.*** Finally, participants also appreciated that these captured snippets served as cognitive scaffolds for their own task performance. P2-4, P6, and P10 reported that being able to look back on what they had seen and thought about helped them better navigate the task, maintain continuity, and build on earlier insights as the task progressed. Rather than merely supporting the system, capturing and revisiting accumulated snippets made users' own reasoning more visible to themselves throughout the task.

3.4 Design Implications: User-Inspectable and Correctable AI Context

While snippet memoing enabled richer and more user-framed context sharing, the study also surfaced a key limitation. As snippets accumulated, participants found it difficult to keep track of what they had captured and how it was reflected in the AI's understanding. This made it harder to revisit prior thoughts, see what had already been considered, and ensure that the AI remained aligned with what they actually found important (P2–P10). Importantly, participants did not simply express a need for better organization of snippets. Rather, they wanted the AI's context itself to reflect their thinking in a way that they could see and influence. At the same time, relying solely on users to manage accumulated context made it difficult to maintain alignment with the AI. Participants instead preferred system support while retaining control over how their context is represented (P6–P8). Based on these observations, we derive two key design implications for supporting the structuring and use of captured context in human–AI collaboration: **DI1. Ground AI context in users' cognitive processes.** Systems should incorporate users' in-situ thoughts as first-class inputs in forming AI context, so that the resulting context reflects what users consider important and how they interpret information. Rather than relying solely on system-inferred signals, this allows the AI's understanding to be directly shaped by the user's perspective. **DI2. Make AI context inspectable and correctable by users.** The AI's context should be externalized in a form that users can directly inspect and revise. This allows users to verify how their inputs are being interpreted, identify mismatches, and adjust the context when needed [35]. By keeping users in the loop, systems can maintain alignment with users' thinking while preserving their sense of authorship and control.

4 Contexty

Based on the design implications, we developed Contexty, a system that externalizes AI context as a user-inspectable and correctable representation (DI2) grounded in users' cognitive processes (DI1). It extends snippet memoing (in Sec 3.1) by incorporating users' in-situ thoughts as first-class inputs to AI context, and surfaces this context through a canvas-based representation that makes it inspectable and correctable by users. The canvas serves both as an inspectable view of the context grounding AI responses and as a workspace where users organize and refine their evolving understanding of the task. Because the context is continuously assembled from users' ongoing task activity and in-situ thoughts, refining the canvas simultaneously refines the AI's grounding. Using Contexty, as users work across different sources, they can capture snippets along with their in-situ thoughts (Figure 1A); Contexty integrates these into the AI's context and surfaces them on the canvas (Figure 1C, D). Users can inspect and refine this canvas at any time, and converse with an AI assistant (Figure 1E) grounded in this context throughout their workflow.

4.1 Context Capture

Contexty captures user context through three complementary channels—*snippet memoing*, *computer-use observations*, and *chat*. Building on prior work showing that passive observation of users' on-screen activity can provide rich contextual signals for both LLMs and computer-use agents [20, 23, 32, 33], Contexty supports **computer-use observation**. Observations provide broad situational awareness of users' ongoing activities without requiring additional effort, enabling the system to capture background task context across applications and over time. Contexty monitors global input events and automatically captures a full-screen screenshot whenever the user clicks or presses `Enter` key outside Contexty. Users can disable passive observation at any time if preferred. **Snippet memoing** follows the same interaction design as the research probe (Section 3.1, Figure 2A). Finally, **chat** exchanges between the user and LLM assistant are also treated as a first-class context channel. Each user prompt and its corresponding AI response are stored alongside snippets and observations as part of the shared task context, allowing future responses to build on the user's prior interactions.

4.1.1 ***Processing and Display.*** For snippets and observations, the system extracts provenance metadata (`appName`, `windowTitle`, `url`) at capture time using OS-level APIs. Each captured item, along with its metadata, is analyzed by an LLM and rendered as a card on the canvas (Figure 4), displaying (A) an AI-generated title, (B) provenance information including the source `appName`, `url`, and timestamp, (C) the raw captured content, (D) AI-generated tags, and (E) an AI-generated user's context reflecting how the system interprets the user's current activity and intent. For snippet captures, the user's in-situ memo is additionally shown alongside the card (F). Cards are color-coded by source type: yellow for snippets, green for observations, and blue for chat. Users can edit any card via an edit button to correct or refine its content. Near-identical observation screenshots are deduplicated via perceptual hashing,

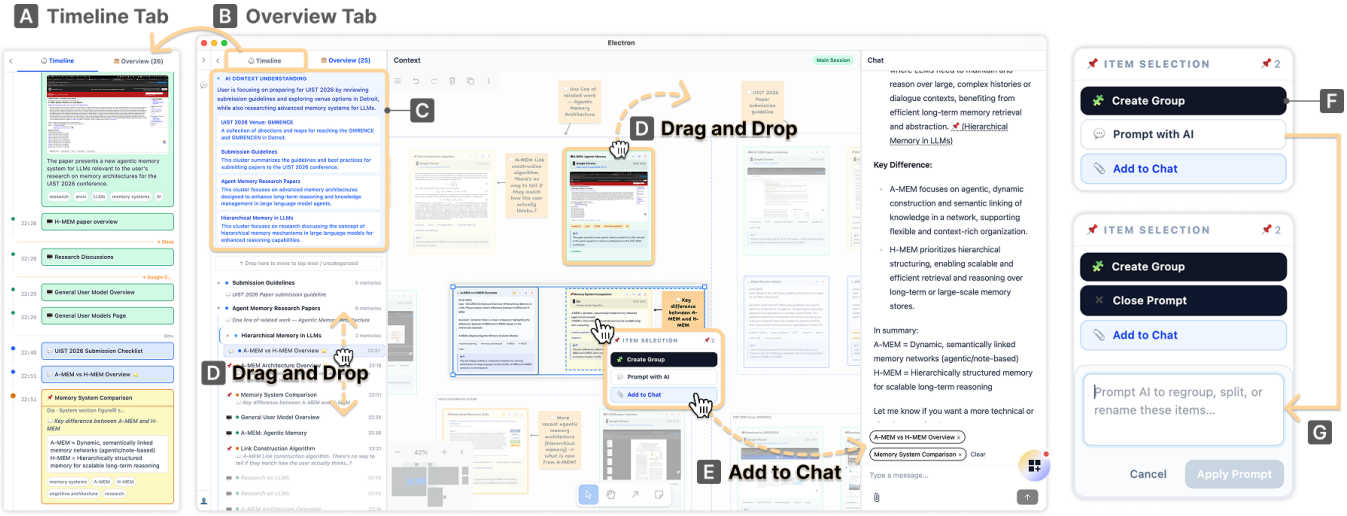

Figure 3: Overview of the main window of Contexty. The memory overview provides both timeline (A) and hierarchical overview (B) views of captured context. The system maintains an AI-generated context summary (C) of the user's current focus. Users can reorganize items and restructure groups via drag-and-drop on the canvas (D), and by right-clicking selected items to group them into new branches or invoke AI-assisted grouping (F, G). Selected items can also be added to chat (E).

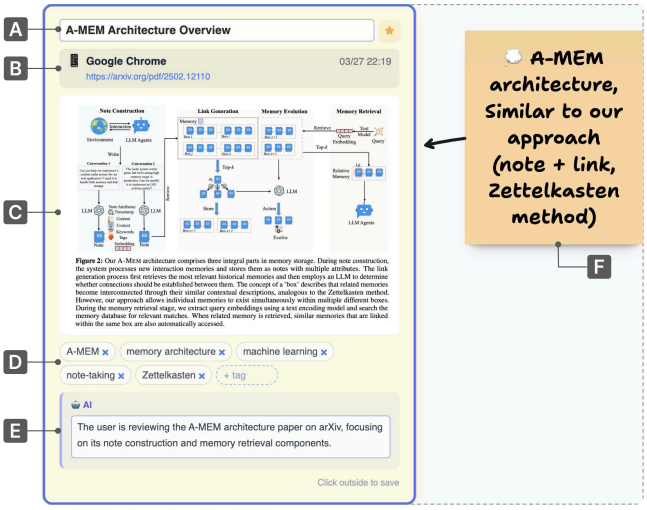

Figure 4: A memory card on the Contexty canvas (edit view). Each card displays (A) an AI-generated title, (B) provenance information including the source application, timestamp, and URL, (C) the raw captured content, (D) AI-generated tags, and (E) an AI-generated user's context reflecting how the system interprets the user's current activity at the time of capture. For snippet captures, (F) the user's in-situ memo is also displayed as a sticky note alongside the card.

and redundant observations are automatically hidden on the canvas based on semantic similarity to existing visible memories (see Appendix B for details).

4.2 Context Memory Structure

As captured items from **observation**, **snippets**, and **chat** accumulate over time, Contexty organizes them into a memory structure to effectively incorporate them into AI's context. Inspired by recent agentic memory architectures, which organize LLM memories into Zettelkasten-inspired note-link structure or hierarchical structure [40, 46], Contexty organizes captured items into a hierarchical memory structure $T = (M, B, L)$, where M is a set of *memories* (individual captured items), B is a set of *branches* (semantic groups of memories with LLM-generated names and summaries), and L is a set of *links* $l = (m_i, m_j, \sigma)$ capturing relationships across memories, each weighted by a relevance score $\sigma \in [0, 1]$. Each memory $m \in M$ is assigned to a branch $b \in B$ (or left unassigned). Each link $l \in L$ is inferred as new memories arrive and is used internally to organize the memory structure.

4.2.1 ***Automatic Placement.*** As new memories are added, the system automatically places each item into the existing structure to maintain a coherent organization of the AI context without requiring manual effort. When a new memory m is added, the system retrieves the top-k semantically related existing items using an LLM, along with relevance scores $\sigma_i \in [0, 1]$, which constitute the links in L. The scores are aggregated at the branch level: $S(b) = \sum_{m_i \in b} \sigma_i$. The memory is assigned to $b^* = \arg\max_{b \in B} S(b)$ if $S(b^*) > 0$; otherwise, a new branch is created from the memory together with its related unassigned items, or the memory is left unassigned if there are none.

4.2.2 ***Memory Evolution.*** As new memories are added, the system continuously refines the existing memory structure to keep it aligned with the user's task context. When a new memory provides additional context, the LLM suggests metadata updates for related existing items, including refined tags and context descriptions. When related memories are updated, an "Updated" badge is

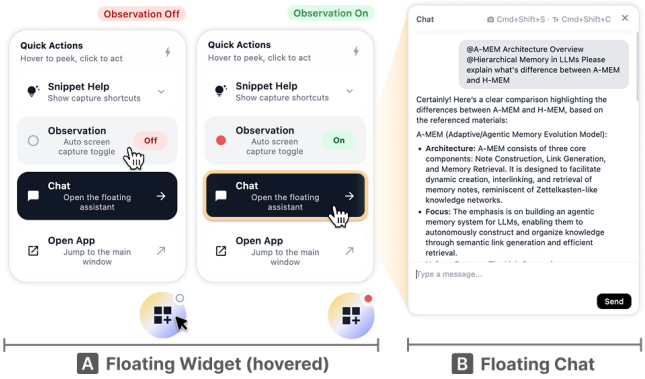

Figure 5: Floating widget (A) provides quick access to snippet memoing, observation toggle, and (B) a mirrored chat.

shown on the affected cards, signaling to the user that the system has revised its interpretation of previously captured content.

4.2.3 ***Users' Reorganization.*** Users can directly modify the memory structure T through multiple interaction methods on the canvas. They can drag-and-drop items (Figure 3D) to move memories (m) across branches (b). Users can also select a set of items and, via right-click, group them into a new branch (Figure 3F). Alternatively, users can provide natural language instructions (e.g., "group these by project") to let the system propose and apply structural updates (Figure 3G).

4.3 Context-Augmented Chat

Contexty provides a chat interface that grounds AI responses in the user's accumulated task context. The interface is available both as a panel fixed alongside the canvas and as a floating widget (Figure 5B) that can be positioned anywhere on the screen, with both views remaining synchronized.

Users can explicitly bring specific memories into a conversation by @-mentioning them or by right-clicking a card on the canvas and selecting "Add to Chat" (Figure 3E). Beyond explicit references, the system automatically retrieves memories relevant to the user's query, taking into account lexical overlap, tag matches, and recency (see Appendix A for details). The retrieved memories are prepended to the user's message with source metadata, allowing the AI to respond with awareness of the user's broader task context.

Contexty also maintains an **AI Context Understanding**, a concise LLM-generated summary of the user's current activities and focus (Figure 3C), derived from the memory structure by analyzing group structures, recent memories weighted by source type (with snippets weighted more heavily than observations), and strong links between memories, which may span branches and indicate concurrent work across domains.

4.4 User Interface

Contexty presents a four-panel layout (Figure 1): (1) a **thread panel** for managing multiple conversation threads and session lifecycle, (2) a **memory overview panel** displaying captured memories either chronologically as a timeline view or as a hierarchical overview of the context memory structure, switchable via tabs, (3) a **canvas panel** providing an interactive spatial visualization of the context memory structure, and (4) a **chat panel** for conversational interaction with the LLM assistant. In addition, a **persistent floating widget** remains visible above all applications regardless of the user's current task. It provides quick access to snippet memoing shortcuts, observation toggling, and a mirrored chat panel so users can interact with the AI without switching back to the system. The canvas renders each branch as a group of memory cards, with child branches displayed as nested groups within their parent—allowing hierarchical task structure to be reflected spatially. Users can reorganize the structure through drag-and-drop on either the canvas or the hierarchical overview panel, moving individual memories or entire groups to reassign their place in the hierarchy.

4.5 Implementation

Contexty is implemented as a cross-platform desktop application using Electron, TypeScript, and React. On the backend, core services handle context capture, memory management, and chat orchestration, with session data persisted locally. On the frontend, the interactive canvas is built on tldraw [42], and the floating widget is rendered as an always-on-top overlay window. LLMs are used throughout the system pipeline, all powered by `gpt-4o-mini-2024-07-18`: content analysis, memory placement, related-item retrieval, and AI-assisted reorganization, except for the main conversational chat, which uses `gpt-4.1-2025-04-14`, orchestrated via OpenAI's Assistants API with persistent threads. All prompts used for LLM interactions are provided in the supplementary material.

5 Evaluation

We conducted a within-subjects user study to understand how a canvas-based context representation and in-situ snippet memoing shape users' experience of collaborating with AI on their task, by comparing Contexty with a baseline condition. Specifically, we investigated the following research questions:

- **RQ1**: Does the canvas-based representation in Contexty help users maintain clearer awareness and control over their evolving task context?
- **RQ2**: Do user-generated in-situ snippets improve the quality of AI assistance during tasks?

Participants completed tasks in two conditions, counterbalanced. In the **Contexty** condition, participants used the full system as described above (Section 4). As the effect of in-situ snippet memoing was already explored in our design exploration (Section 3), we designed the baseline to isolate the effect of the canvas representation while retaining context capture. In the **Baseline** condition, the canvas was removed—participants could not view the underlying memory structure or the canvas overview tab. The timeline view, snippet memoing, computer-use observation, and context-augmented chat (including @-mention of memory items) remained available. This baseline approximates how users interact with existing LLM services (e.g., ChatGPT Atlas [28]), extended with snippet memoing and passive context capture. This setup allows us to further assess the effect of snippet memoing during task sessions in both conditions, while isolating the additional contribution of the canvas.

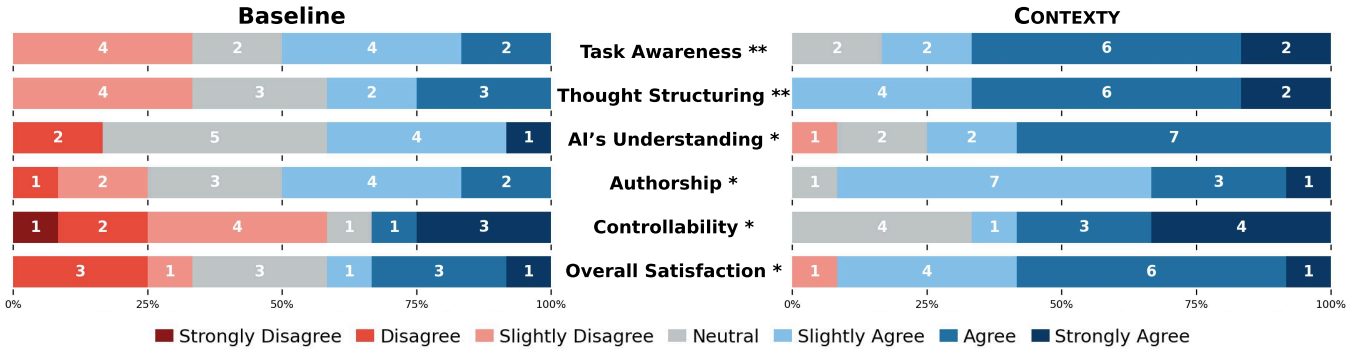

Figure 6: Post-condition survey results comparing Contexty and the baseline across six dimensions (7-point Likert scale). Asterisks denote statistical significance from Wilcoxon signed-rank tests (*$p < .05$, **$p < .01$)

5.1 Participants and Tasks

We recruited 12 participants (5 female, 7 male; age $M = 26.0$, $SD = 3.28$) through online recruitment postings on our university community platform. All participants had prior experience using LLMs (e.g., ChatGPT, Claude, Gemini) for exploratory or sensemaking tasks. Participants were compensated KRW 40,000 (≈ USD 28) for approximately two hours of their time.

To ensure comparable levels of familiarity, motivation, and initial cognitive engagement across participants, we provided three categories of exploratory sensemaking tasks: career-related tasks (e.g., exploring PhD labs, searching for internships), academic and research-related tasks (e.g., surveying related work, organizing research directions), and everyday life tasks (e.g., trip planning, product comparison). In a pre-study survey, participants rated their familiarity with each category on a 5-point Likert scale and selected the two categories they felt most familiar with. To keep the two sessions comparable, we included only participants whose familiarity ratings for the two selected categories differed by at most one point, so that neither condition was paired with a substantially more familiar task. They then brought one real-world task from each selected category that they were genuinely motivated to pursue. Rather than assigning a fixed task, we adopted this open task design for ecological validity, so that participants worked on tasks of genuine personal relevance and brought their own criteria and prior knowledge to bear.

5.2 Procedure

Each session lasted approximately two hours. The session began with a **10-minute introduction**, during which the study was explained, and participants provided informed consent. Participants then completed **two task sessions** (45 minutes each), separated by a short break. In each session, participants first completed a 10-minute system walkthrough with a warm-up task to familiarize themselves with the interface, conducted the task work under the assigned condition for 30 minutes, and responded to a 5-minute post-condition survey. The session concluded with a **15-minute semi-structured interview** to explore participants' subjective experiences across both conditions.

5.3 Measures

5.3.1 ***Effect of the Canvas (RQ1).*** To evaluate the effect of the canvas, we used a post-condition survey after each condition with 7-point Likert scales. The survey captured six dimensions related to user experience and perceived control: *Task Awareness*, *Thought Structuring*, *Perceived AI's Understanding*, *Authorship*, *Controllability*, and *Overall Satisfaction*. We then compared responses between conditions using Wilcoxon signed-rank tests. The exact wording of all questionnaire items is provided in Appendix D.

5.3.2 ***Effect of Snippets (RQ2).*** To assess whether in-situ snippet memoing improves AI response quality, we employed an **in-situ preference probe** during task sessions in both baseline and Contexty conditions. For each user query, the system silently generated two candidate responses in parallel: one using the full context, and one with all snippet-derived context filtered out. To avoid disrupting task flow, the rating was triggered only when the two contexts and their resulting responses were sufficiently different, determined by a two-stage gating mechanism based on context similarity and response-level divergence (see Appendix C for details). When triggered, participants were shown the two responses in randomized order without knowing whether each response was generated with or without snippet-derived context. They selected their preferred response and could optionally provide a brief reason for their choice, which was further discussed during the post-task interview. We report the selection ratio of snippet-informed responses as an indicator of snippet utility.

6 Results

Contexty significantly outperformed the baseline across all six dimensions (Figure 6). Beyond these overall improvements, participants reported better task awareness, stronger authorship and control, and improved alignment with the AI. They also engaged with the canvas through both active refinement and passive inspection, while snippet memoing improved response quality and supported more intentional context sharing. We report the findings organized into themes based on quantitative and interview data.

6.1 RQ1: Effect of the Canvas on Task Awareness and Sense of Control

6.1.1 ***Enhancing Task Awareness and Reinforcing Authorship.*** The canvas significantly improved participants' *task awareness* (M = 5.67 vs. 4.33, Δ = +1.33, W = 0.0, p = .0039**) and *authorship* (M = 5.33 vs. 4.33, Δ = +1.00, W = 2.0, p = .0312*). Contexty not only helped users maintain awareness of their ongoing task, but also supported a stronger sense that the task reflected their own thinking rather than the AI's suggestions. Participants described the canvas as making their task progress visible and traceable, enabling them to revisit prior steps without mentally reconstructing them (P01-8, P10, P12). Importantly, this visibility contributed to a sense of ownership. Participants (P02, P05, P12) described the canvas not as an AI-managed memory, but as their own space for maintaining and shaping task context. As P12 noted, *"gave me a sense of ownership over the task, this in contrast to the conventional LLM systems where outputs often feel more like the AI's product than my own."*

6.1.2 ***Supporting Thought Structuring Through Both Active Organization and Passive Inspection.*** Participants reported significantly higher *thought structuring* with Contexty (M = 5.83 vs. 4.33, Δ = +1.50, W = 0.0, p = .0039**), indicating that the canvas effectively supported organization of information and ideas. Importantly, this effect emerged regardless of the usage patterns. Some participants (P02, P03, P05, P10-12) actively reorganized and edited canvas items, while others (P04, P06, P09) primarily relied on passively inspecting the AI-generated organization (Figure 11 in the Appendix). These suggest two complementary pathways for supporting thought structuring: ***active organization*** and ***passive inspection***. Participants who engaged in ***active reorganization*** treated the AI-generated organization as a starting point, editing and regrouping items to reflect their evolving understanding. P03 noted that reorganizing helped them identify what to explore next or what additional information to seek, while P12 described refining the structure as a way to clarify their own thinking as the task progressed. In contrast, participants who engaged in ***passive inspection*** used the canvas as a scaffold for understanding the current state of their task, glancing at the overview to reorient themselves and then returning to their task. P09 noted, "I focused more on getting answers than reorganizing the canvas, yet when I briefly glanced at it, I felt my task context was fairly well structured."

6.1.3 ***Fostering Trust and Alignment Through Visibility into AI's Understanding.*** Participants reported significantly higher *perceived AI's understanding* in Contexty than in the baseline (M = 5.25 vs. 4.25, Δ = +1.00, W = 3.5, p = .0273*), suggesting that the canvas helped participants feel that the AI better understood their context and intentions during the task. This leads to increased trust and alignment with the AI. By making the AI's interpretation of users' context visible and editable, users could inspect how their inputs were being interpreted and correct misalignments. P01 described that observing how the AI organizes the information increased their trust. P10 noted that being able to see what the AI was referencing made it easier to verify and correct its understanding when needed.

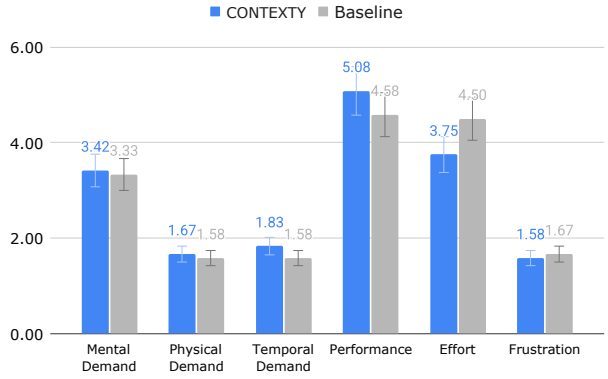

Figure 7: Comparison of NASA-TLX ratings between Contexty and Baseline.

6.1.4 ***Increased Controllability while Introducing a Tension with Curation Effort and Noise.*** Participants reported significantly higher *controllability* when using Contexty (M = 5.58 vs. 4.00, Δ = +1.58, W = 6.0, p = .0332*), suggesting that the canvas allowed users to actively intervene in, modify, and curate their task context rather than relying solely on the AI's automatic organization. However, qualitative findings reveal an important tension. While the canvas increased controllability, it also introduced the need for users to curate and filter the accumulated context. Because the system continuously captured information through automatic observation, participants often encountered irrelevant or low-value items, particularly in multitasking scenarios. Although it contributed to richer context capture, participants frequently reported that it introduced unnecessary information, like noise, which increases curation effort (P10, P12). P10 even expressed a preference not to use automatic observation to maintain a cleaner task context. Interestingly, this additional curation effort was not reflected in the NASA-TLX results, where no significant differences were observed (Figure 7). While Contexty introduces curation effort, the baseline required users to mentally organize and keep track of relationships among captured items using only a chronological list. The two conditions therefore differed in the form of effort rather than its overall amount, which may explain the comparable workload ratings.

6.2 RQ2: Effect of In-situ Snippet Memoing on AI Response Quality

6.2.1 ***With-Snippet Responses Were Consistently Preferred.*** Across all 12 participants, with-snippet responses were preferred at a rate of 78.1% on average (Figure 8). On average, 8.8 out of 18.1 total queries per participant triggered the comparison (48%), and 6.8 of those resulted in selection of the with-snippet response. When explaining their choices, participants reported that the with-snippet responses felt more directly grounded in their own context and perspective. P02 noted that the with-snippet response was clearly better at directly answering what they had asked, and P03 described how the with-snippet response used vocabulary and framing that matched their own way of thinking about the task. However, the preference for with-snippet responses was not always the case. In some cases, participants favored responses generated

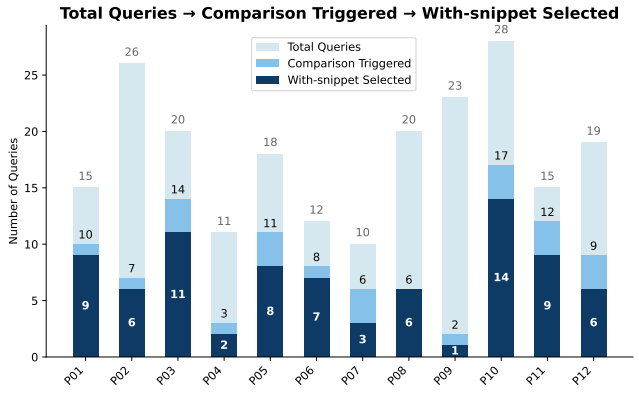

Figure 8: Per-participant query funnel (total queries, comparisons triggered, and with-snippet responses selected). On average, 48% of queries triggered a comparison, and among those, 78% resulted in selection of the with-snippet response.

without snippets as they provided more diverse perspectives or introduced ideas beyond their immediate context. This suggests that while snippets reliably improve response relevance, they may also narrow the response space, leading users to occasionally prefer less constrained answers, especially for ideation or exploration.

6.2.2 ***Snippet Memoing Integrates Naturally into Task Flow, and the Canvas Amplifies Its Effect.*** Interaction log analysis shows that snippet-memoing events occurred steadily throughout sessions (Figure 11 in the Appendix), indicating that it naturally integrated into users' task flow. Participants stated that the snippet memoing, as a lightweight action, did not require switching context or interrupting ongoing work. Notably, several participants reported that the act of writing a snippet memo itself helped them think (P01, P03-9, P12). P01 described how capturing a snippet sometimes steered their thinking in a new direction, and P03 noted that the process of articulating why something mattered helped them clarify what to explore next. Participants reported that their use of snippet memoing differed depending on the presence of the canvas. In the baseline, memos were often brief, summary-like labels primarily for later retrieval. In contrast, with Contexty, the canvas enabled these snippets to be organized, which led participants to record richer thoughts, including connections between pieces of information and personal interpretations (P03–4, P06–7, P11). Participants emphasized that snippets alone were not enough; the canvas gave them structure and made them retrievable and interpretable. As P03 noted, "snippets let me express how I interpret the information, but the canvas is what made those interpretations visible and usable."

6.2.3 ***Trade-offs Between Passive Computer-Use Observation and Snippet Memoing.*** While passive observation automatically captured background task context [32], participants found it insufficient for conveying their intent and interpretation. As P05 noted, screen captures reflected what they were looking at, but not **why** it mattered or **how** they were making sense of it. In contrast, snippets allowed users to actively shape how their context was shared with the AI. Participants noted that this helped them communicate what they considered important (P03), such as connecting new information to prior knowledge or providing examples to clarify intent (P07, P10). At the same time, participants acknowledged that continuous observation could capture broader context and even retain information they might otherwise miss. However, these benefits were often outweighed by concerns about noise and unintended information. Notably, no participant disabled passive observation during the study, though several remarked in the post-task interview that they would prefer limiting or turning off observation, noting that while they could tell when to turn it off for privacy, it was less clear what made it worth keeping on. Together, these findings suggest that while observation provides useful background context, snippet memoing serves as a more intentional, user-controllable channel for sharing their task context that matters more for AI assistance.

7 Discussion

Building on our findings, we discuss (1) how systems can capture user context while minimizing user burden, (2) how such context can be represented in forms that are understandable by both users and AI, and (3) how such an approach can extend beyond single sessions to support sustainable context use over time.

Our findings reveal a tension between comprehensive context capture and selective control. Passive observation contributed to a richer context by capturing information users might otherwise miss, but also introduced noise and an additional curation burden, particularly in multitasking scenarios. At the same time, relying solely on user-driven externalization places a burden on the user. Balancing these two modalities is therefore an important design challenge. One promising approach to navigating this balance is to let the system learn from users' behavior. By leveraging snippet memoing patterns—what and where users choose to capture, and the thoughts they attach as memos—alongside patterns of context items actively referenced in chat, the system can infer which types of information tend to matter to an individual user. This inference could then drive proactive surfacing of relevant signals from observed behavior, such as highlighting passages on pages similar to previously snippeted content or flagging new information related to concepts the user frequently referenced in their memos. However, as agents take a more active role in shaping and surfacing such context, there is a risk that AI-generated organizations may inadvertently steer how users understand and pursue their tasks. Ensuring that such support remains transparent and correctable by the user will be an important consideration for future work.

In human teamwork, effective collaboration depends on building and maintaining a *shared mental model* of the task, including each other's goals, priorities, and evolving understanding [4, 25]. Our findings suggest a similar need in human–AI collaboration: making task context shared and inspectable improved users' sense of authorship, control, and perceived AI understanding of the task. Yet this raises an important open question: what should such a shared representation actually look like? Humans benefit from spatial, visual organization that supports reflection and navigation [5, 34]. AI agents, by contrast, may need more structured textual representations that can be incorporated into model inputs, such as `llms.txt` [12], which encodes complex content in LLM-readable

format. This suggests the need for a translation layer between human-facing and AI-facing representations of context.

Contexty takes an initial step in this direction by externalizing the AI's context on a canvas that users can inspect and revise. The canvas presents context in an intuitive, spatial form, while grounding it in the underlying hierarchical memory structure used by the AI. In this way, the canvas functions as a translation layer that makes AI context accessible to users, while also allowing users' cognitive processes to be incorporated. This approach is facilitated by the form of AI context used in our system, which organizes LLM memories into a hierarchical structure inspired by recent LLM memory work [40, 46]. However, as AI context representations evolve, they may not always align with formats that are easily interpretable or editable by users. Future work should therefore explore how such translation layers can generalize beyond specific representations, and how to maintain alignment between human-meaningful context and model-effective representations as both continue to evolve.

The synergy between in-situ snippet capturing and canvas-based organization also points to a broader design opportunity of embedding cognitive trail into everyday work environments. Recent AI-powered browsers have begun to integrate AI into users' ongoing knowledge work, supporting activities such as information gathering and assistance within the browsing experience [6, 14, 28, 30]. Our findings suggest that these systems could benefit from making users' cognitive processes explicit and inspectable. Rather than leaving them implicitly inferred by AI or transient, a canvas-like layer could make users' cognitive trail visible and continuously usable within their primary work environment.

Finally, the notion of user-inspectable and correctable AI context, grounded in users' cognitive traces, carries important implications beyond single-session interactions. As this context accumulates over time, it calls for more systematic approaches to managing and reusing memory. This includes distinguishing between short-term and long-term context [10, 39], organizing information at the task or project level, and supporting selective reuse of relevant past context across sessions. This opens up opportunities for more sustainable interaction, where past cognitive traces can be maintained, refined, and brought forward when needed. To support this, future work should explore mechanisms for filtering and prioritizing accumulated context, helping users decide what to retain, revisit, or discard over time. This includes developing representations and interaction techniques that allow users to selectively surface relevant past context without being overwhelmed, while preserving their ability to inspect and correct how it is used by the AI.

8 Limitations

Our study focuses on single-session tasks, limiting our understanding of how context-sharing behaviors evolve over time; future work should examine how users manage and reuse accumulated context across sessions through longitudinal deployment. The tasks we studied primarily involved information foraging and sensemaking, leaving open how snippet usage and canvas-based organization may differ in artifact-producing tasks such as writing or programming. Finally, although Contexty externalizes AI context in a user-inspectable and correctable form, it is grounded in a hierarchical memory structure, which may limit its generalizability across tasks and longer-term use.

9 Conclusion

We presented Contexty, a system that incorporates users' in-situ thoughts into AI context and makes it visible and editable in a shared space for human–AI collaboration. Our evaluation showed that Contexty improves users' task awareness, sense of authorship, control, and perceived AI understanding. Our work highlights the importance of moving beyond implicitly inferred, system-managed context toward user-inspectable and correctable AI context that incorporates users' cognitive processes. We hope this work motivates future systems that support more transparent and collaborative ways of constructing context, keeping humans and AI aligned in complex tasks.

Acknowledgments

We appreciate the reviewers for their valuable feedback and comments. We also thank the members of KIXLAB for their help in polishing this manuscript. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) (No.RS-2025-00557726). This work was also supported by the Office of Naval Research (ONR: N00014-24-1-2290).

References

[1] Sönke Ahrens. 2022. *How to take smart notes: One simple technique to boost writing, learning and thinking.* Sönke Ahrens.

[2] Anthropic. 2025. How Claude Code works - Claude Code Docs. https://code.claude.com/docs/en/how-claude-code-works [Accessed 2026-03-31].

[3] Stephen Brade, Bryan Wang, Mauricio Sousa, Sageev Oore, and Tovi Grossman. 2023. Promptify: Text-to-image generation through interactive prompt exploration with large language models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–14.

[4] Janis A Cannon-Bowers, Eduardo Salas, and Sharolyn Converse. 1993. Shared mental models in expert team decision making. (1993).

[5] James M Clark and Allan Paivio. 1991. Dual coding theory and education. *Educational psychology review* 3, 3 (1991), 149–210.

[6] Dia. [n. d.]. Dia Browser | AI Chat With Your Tabs. https://www.diabrowser.com/ [Accessed 2026-03-31].

[7] Electron. [n. d.]. Build cross-platform desktop apps with JavaScript, HTML, and CSS | Electron. https://www.electronjs.org/ "[Accessed 2026-03-31]".

[8] Frederic Gmeiner, Nicolai Marquardt, Michael Bentley, Hugo Romat, Michel Pahud, David Brown, Asta Roseway, Nikolas Martelaro, Kenneth Holstein, Ken Hinckley, and Nathalie Riche. 2025. Intent Tagging: Exploring Micro-Prompting Interactions for Supporting Granular Human-GenAI Co-Creation Workflows. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 531, 31 pages. https://doi.org/10.1145/3706598.3713861

[9] Frederic Gmeiner, John Thompson, George Fitzmaurice, and Justin Matejka. 2026. PointAloud: An Interaction Suite for AI-Supported Pointer-Centric Think-Aloud Computing. *arXiv preprint arXiv:2602.09296* (2026).

[10] Kostas Hatalis, Despina Christou, Joshua Myers, Steven Jones, Keith Lambert, Adam Amos-Binks, Zohreh Dannenhauer, and Dustin Dannenhauer. 2023. Memory matters: The need to improve long-term memory in llm-agents. In *Proceedings of the AAAI Symposium Series*, Vol. 2. 277–280.

[11] James Hollan, Edwin Hutchins, and David Kirsh. 2000. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Trans. Comput.-Hum. Interact.* 7, 2 (June 2000), 174–196. https://doi.org/10.1145/353485.353487

[12] Jeremy Howard. [n. d.]. The /llms.txt file – llms-txt. https://llmstxt.org/ [Accessed 2026-03-31].

[13] Peiling Jiang, Jude Rayan, Steven P Dow, and Haijun Xia. 2023. Graphologue: Exploring large language model responses with interactive diagrams. In *Proceedings of the 36th annual ACM symposium on user interface software and technology*. 1–20.

[14] Peiling Jiang and Haijun Xia. 2025. Orca: Browsing at scale through user-driven and ai-facilitated orchestration across malleable webpages. *arXiv preprint arXiv:2505.22831* (2025).

[15] Hyeonsu B Kang, Tongshuang Wu, Joseph Chee Chang, and Aniket Kittur. 2023. Synergi: A Mixed-Initiative System for Scholarly Synthesis and Sensemaking. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 43, 19 pages. https://doi.org/10.1145/3586183.3606759

[16] Yoonsu Kim, Kihoon Son, Seoyoung Kim, Brandon Chin, and Juho Kim. 2026. IntentFlow: Investigating Fluid Dynamics of Intent Communication in Generative AI. In *Proceedings of the 2026 Designing Interactive Systems Conference* (Singapore, Singapore) *(DIS '26)*. Association for Computing Machinery, New York, NY, USA, 3804–3837. https://doi.org/10.1145/3800645.3812999

[17] Rebecca Krosnick, Fraser Anderson, Justin Matejka, Steve Oney, Walter S. Lasecki, Tovi Grossman, and George Fitzmaurice. 2021. Think-aloud computing: Supporting rich and low-effort knowledge capture. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–13.

[18] Andrew Kuznetsov, Joseph Chee Chang, Nathan Hahn, Napol Rachatasumrit, Bradley Breneisen, Julina Coupland, and Aniket Kittur. 2022. Fuse: In-Situ sensemaking support in the browser. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–15.

[19] Michelle S Lam, Fred Hohman, Dominik Moritz, Jeffrey P Bigham, Kenneth Holstein, and Mary Beth Kery. 2025. Policy Maps: Tools for Guiding the Unbounded Space of LLM Behaviors. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–24.

[20] Michelle S Lam, Omar Shaikh, Hallie Xu, Alice Guo, Diyi Yang, Jeffrey Heer, James A Landay, and Michael S Bernstein. 2025. Just-In-Time Objectives: A General Approach for Specialized AI Interactions. *arXiv preprint arXiv:2510.14591* (2025).

[21] Sangwook Lee, Adnan Abbas, Yan Chen, Young-Ho Kim, and Sang Won Lee. 2025. CHOIR: A Chatbot-mediated Organizational Memory Leveraging Communication in University Research Labs. *arXiv preprint arXiv:2509.20512* (2025).

[22] Wengxi Li, Jingze Tian, and Can Liu. 2026. Orality: A Semantic Canvas for Externalizing and Clarifying Thoughts with Speech. *arXiv preprint arXiv:2603.02544* (2026).

[23] Yaxi Lu, Shenzhi Yang, Cheng Qian, Guirong Chen, Qinyu Luo, Yesai Wu, Huadong Wang, Xin Cong, Zhong Zhang, Yankai Lin, et al. 2024. Proactive agent: Shifting llm agents from reactive responses to active assistance. *arXiv preprint arXiv:2410.12361* (2024).

[24] Catherine C Marshall. 1997. Annotation: from paper books to the digital library. In *Proceedings of the second ACM international conference on Digital libraries*. 131–140.

[25] John E Mathieu, Tonia S Heffner, Gerald F Goodwin, Eduardo Salas, and Janis A Cannon-Bowers. 2000. The influence of shared mental models on team process and performance. *Journal of applied psychology* 85, 2 (2000), 273.

[26] Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, et al. 2025. A survey of context engineering for large language models. *arXiv preprint arXiv:2507.13334* (2025).

[27] Donald Norman. 1991. *Cognitive Artifacts*. 333.

[28] OpenAI. [n. d.]. ChatGPT Atlas. https://chatgpt.com/atlas/ [Accessed 2026-03-31].

[29] OpenAI. 2024. What is Memory? - OpenAI Help Center. https://help.openai.com/en/articles/8983136-what-is-memory [Accessed 2026-03-31].

[30] Perplexity. [n. d.]. Comet Browser: a Personal AI Assistant. https://www.perplexity.ai/comet [Accessed 2026-03-31].

[31] Donald A Schön. 1992. *The reflective practitioner: How professionals think in action*. Routledge. https://doi.org/10.4324/9781315237473

[32] Omar Shaikh, Shardul Sapkota, Shan Rizvi, Eric Horvitz, Joon Sung Park, Diyi Yang, and Michael S. Bernstein. 2025. Creating General User Models from Computer Use. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 35, 23 pages. https://doi.org/10.1145/3746059.3747722

[33] Omar Shaikh, Valentin Teutschbein, Kanishk Gandhi, Yikun Chi, Nick Haber, Thomas Robinson, Nilam Ram, Byron Reeves, Sherry Yang, Michael S. Bernstein, and Diyi Yang. 2026. Learning Next Action Predictors from Human-Computer Interaction. arXiv:2603.05923 [cs.CL] https://arxiv.org/abs/2603.05923

[34] Frank M Shipman III and Catherine C Marshall. 1999. Spatial hypertext: an alternative to navigational and semantic links. *ACM Computing Surveys (CSUR)* 31, 4es (1999), 14–es.

[35] Ben Shneiderman. 1983. Direct manipulation: A step beyond programming languages. *Computer* 16, 08 (1983), 57–69.

[36] Kihoon Son, DaEun Choi, Tae Soo Kim, Young-Ho Kim, Sangdoo Yun, and Juho Kim. 2025. ClearFairy: Capturing Creative Workflows through Decision Structuring, In-Situ Questioning, and Rationale Inference. *arXiv preprint arXiv:2509.14537* (2025).

[37] Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. 2024. Luminate: Structured generation and exploration of design space with large language models for human-ai co-creation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–26.

[38] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. 2023. Sensecape: Enabling multilevel exploration and sensemaking with large language models. In *Proceedings of the 36th annual ACM symposium on user interface software and technology*. 1–18.

[39] Theodore Sumers, Shunyu Yao, Karthik R Narasimhan, and Thomas L Griffiths. 2023. Cognitive architectures for language agents. *Transactions on Machine Learning Research* (2023).

[40] Haoran Sun, Shaoning Zeng, and Bob Zhang. 2026. H-MEM: Hierarchical Memory for High-Efficiency Long-Term Reasoning in LLM Agents. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. 341–350.

[41] Craig S. Tashman and W. Keith Edwards. 2011. LiquidText: a flexible, multitouch environment to support active reading. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 3285–3294. https://doi.org/10.1145/1978942.1979430

[42] tldraw. [n. d.]. tldraw: Infinite Canvas SDK for React. https://tldraw.dev/ "[Accessed 2026-03-31]".

[43] Priyan Vaithilingam, Munyeong Kim, Frida-Cecilia Acosta-Parenteau, Daniel Lee, Amine Mhedhbi, Elena L. Glassman, and Ian Arawjo. 2025. Semantic Commit: Helping Users Update Intent Specifications for AI Memory at Scale. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 137, 18 pages. https://doi.org/10.1145/3746059.3747778

[44] Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. 2025. OpenHands: An Open Platform for AI Software Developers as Generalist Agents. arXiv:2407.16741 [cs.SE] https://arxiv.org/abs/2407.16741

[45] Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. 2024. PromptCharm: Text-to-Image Generation through Multi-modal Prompting and Refinement. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 185, 21 pages. https://doi.org/10.1145/3613904.3642803

[46] Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110* (2025).

[47] Bian Yang, Fan Gu, and Xiamu Niu. 2006. Block mean value based image perceptual hashing. In *2006 International Conference on Intelligent Information Hiding and Multimedia*. IEEE, 167–172.

[48] Bufang Yang, Lilin Xu, Liekang Zeng, Kaiwei Liu, Siyang Jiang, Wenrui Lu, Hongkai Chen, Xiaofan Jiang, Guoliang Xing, and Zhenyu Yan. 2025. Contextagent: Context-aware proactive llm agents with open-world sensory perceptions. *arXiv preprint arXiv:2505.14668* (2025).

[49] Saelyne Yang, Jaesang Yu, Yi-Hao Peng, Kevin Qinghong Lin, Jae Won Cho, Yale Song, and Juho Kim. 2026. GUIDE: A Benchmark for Understanding and Assisting Users in Open-Ended GUI Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. To appear.

[50] Runlong Ye, Patrick Lee, Matthew Varona, Oliver Huang, and Carolina Nobre. 2025. ScholarMate: A Mixed-Initiative Tool for Qualitative Knowledge Work and Information Sensemaking. In *Adjunct Proceedings of the 4th Annual Symposium on Human-Computer Interaction for Work (CHIWORK '25 Adjunct)*. Association for Computing Machinery, New York, NY, USA, Article 7, 7 pages. https://doi.org/10.1145/3707640.3731913

[51] Ryan Yen and Jian Zhao. 2024. Memolet: Reifying the Reuse of User-AI Conversational Memories. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 58, 22 pages. https://doi.org/10.1145/3654777.3676388

[52] Jiajie Zhang. 1997. The nature of external representations in problem solving. *Cognitive science* 21, 2 (1997), 179–217.

[53] Xiaoyu Zhang, Senthil Chandrasegaran, and Kwan-Liu Ma. 2021. Conceptscope: Organizing and visualizing knowledge in documents based on domain ontology. In *Proceedings of the 2021 chi conference on human factors in computing systems*. 1–13.

[54] Xiaoyu Zhang, Jianping Li, Po-Wei Chi, Senthil Chandrasegaran, and Kwan-Liu Ma. 2023. ConceptEVA: Concept-based interactive exploration and customization of document summaries. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–16.

[55] Zheng Zhang, Weirui Peng, Xinyue Chen, Luke Cao, and Toby Jia-Jun Li. 2025. LADICA: a large shared display interface for generative AI cognitive assistance in co-located team collaboration. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–22.

[56] Dora Zhao, Diyi Yang, and Michael S Bernstein. 2025. Knoll: Creating a knowledge ecosystem for large language models. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–23.

A Context Retrieval for Chat

When a user sends a query q, the system retrieves relevant memories in two stages.

Explicit References. Any memories or branches explicitly referenced by the user (via @-mentions or "Add to Chat") are resolved first. For branch references, the system collects the branch's member memories sorted by recency and selects the most recent items as representatives.

Automatic Retrieval. The remaining non-archived memories are scored against the query using a composite ranking function:

$$\begin{aligned}\text{score}(m, q) = &\underbrace{\frac{|\text{tokens}(q) \cap \text{tokens}(m)|}{|\text{tokens}(q)|}}_{\text{token overlap}} \\ &+ \underbrace{0.2 \cdot \mathbf{1}[\exists\, t \in \text{tags}(m) : t \subseteq q]}_{\text{tag boost}} \\ &+ \underbrace{0.15 \cdot \max\left(0,\ 1 - \frac{\text{age}(m)}{30\text{d}}\right)}_{\text{recency}}\end{aligned}$$

Here, $\text{tokens}(m)$ is derived from the concatenation of the memory's title, description, context, intent annotations, raw text, tags, and provenance metadata (application name, window title, URL). The tag boost rewards memories whose tags appear as substrings in the query, and the recency term decays linearly over a 30-day window.

Merging and Formatting. Explicitly referenced memories take priority; automatically retrieved memories fill the remaining slots. Each selected memory is formatted as a structured text block containing its source type, title, context, summary, and tags, and prepended to the user's message. Explicitly referenced items are flagged as `MENTION` to signal higher priority to the LLM.

B Observation Filtering Pipeline

Passive observation can produce a high volume of captures, many of which are visually or semantically redundant. Contexty applies a two-stage filtering pipeline to manage this.

Stage 1: Perceptual Hash Deduplication. Before any LLM analysis, each observation screenshot is converted to a perceptual hash using the BMVBHash algorithm [47]. The image is resized to 16×16 grayscale and hashed into a 256-bit vector. The system computes the Hamming distance d_H between the new hash and all previously accepted hashes:

$$d_H(h_{\text{new}}, h_i) = \sum_{j=1}^{256} \mathbf{1}[h_{\text{new}}^{(j)} \neq h_i^{(j)}]$$

If $d_H \leq 10$ for any existing hash, the screenshot is considered a near-duplicate and discarded before further processing. This lightweight check eliminates visually identical or near-identical captures (e.g., repeated screenshots of the same static page) without incurring LLM costs.

Stage 2: Semantic Similarity Auto-Hide. Screenshots that pass the perceptual hash check proceed through the standard analysis pipeline and are added to the memory tree. The system then evaluates whether the new observation is semantically redundant with respect to currently visible memories. Using the related-item retrieval mechanism, the system obtains the top-k related items among the 30 most recent memories, each with an LLM-generated relevance score $\sigma_i \in [0, 1]$. Let $\sigma^* = \max_i \sigma_i$ be the highest score among non-archived, non-hidden memories with source type *observation* or *snippet*. If $\sigma^* \geq 0.65$, the new observation is marked as hidden on the canvas. The memory is retained in the tree and remains available for context retrieval, but is not rendered on the canvas to reduce visual clutter.

This two-stage design is intentional: Stage 1 operates at the pixel level with negligible latency, filtering exact or near-exact recaptures. Stage 2 operates at the semantic level via LLM, catching cases where a user returns to the same task or topic in a different visual state (e.g., scrolling further on the same documentation page). Together, the two stages ensure that the canvas remains manageable while preserving the full capture history for retrieval.

C Preference Probe Implementation Details

Context Construction. For each user query q, the system constructed two context variants in parallel: context C_A, built from the full memory tree (chat history + observations + snippets), and context C_B, built from the same tree but with all snippet-sourced memories and their associated intent annotations filtered out. Each context was independently scored and ranked using the retrieval function described in Appendix A, ensuring that both variants selected their respective top memories under the same ranking criterion.

Stage 1: Context-Level Gate. To filter out cases where the two context variants were effectively equivalent, the system compared C_A and C_B using memory-level Jaccard similarity J_{mem} (over selected memory IDs) and token-level Jaccard similarity J_{tok} (over context text tokens). If $J_{\text{mem}} \geq 0.85$ and $J_{\text{tok}} \geq 0.92$, the contexts were deemed equivalent and only response r_A was shown to the participant.

Stage 2: Response-Level Gate. When the contexts passed Stage 1, both were sent to the same LLM to generate candidate responses r_A and r_B in parallel. The system then applied a final gate based on Normalized Compression Distance (NCD) to determine whether the two responses were substantively different:

$$\text{NCD}(r_A, r_B) = \frac{|r_A \oplus r_B|_z - \min(|r_A|_z, |r_B|_z)}{\max(|r_A|_z, |r_B|_z)}$$

where $|\cdot|_z$ denotes the gzip-compressed byte length and $\oplus$ denotes string concatenation. Only when $\text{NCD}(r_A, r_B) > \tau$ were both responses presented to the participant.

Threshold Calibration. We set $\tau = 0.7$ based on a calibration analysis conducted on 51 response pairs collected from 10 participants during design exploration sessions (Section 3) prior to the main study. For each pair, we computed three similarity metrics—NCD, ROUGE-L, and Jaccard—and obtained an independent LLM judgment of whether the pair exhibited a substantive semantic difference. Figure 9 shows the distribution of each metric across all pairs. To assess discriminative power, Figure 10 splits each metric by the substantive-difference judgment. NCD exhibited the strongest

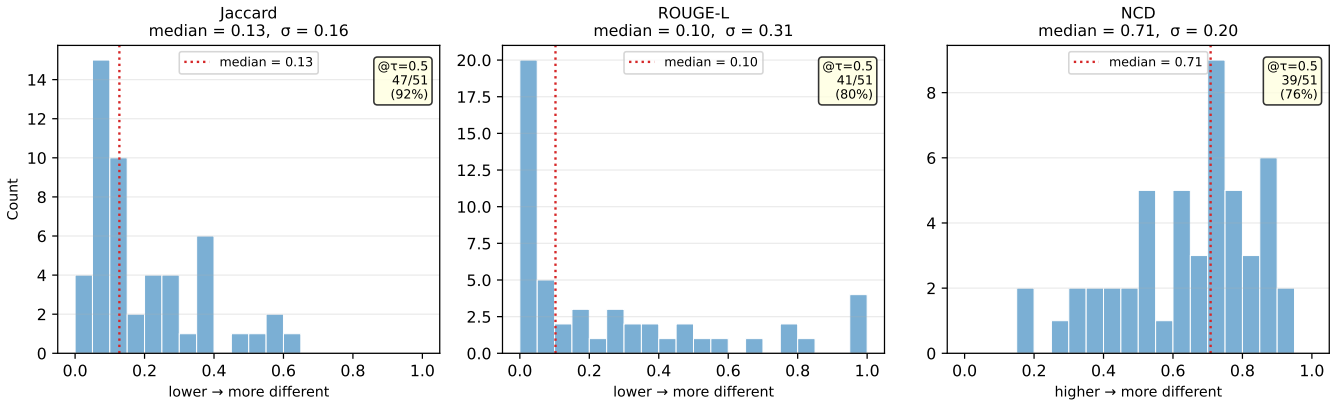

Figure 9: Distribution of three similarity metrics across 51 response pairs from design exploration sessions. Dashed red lines indicate medians.

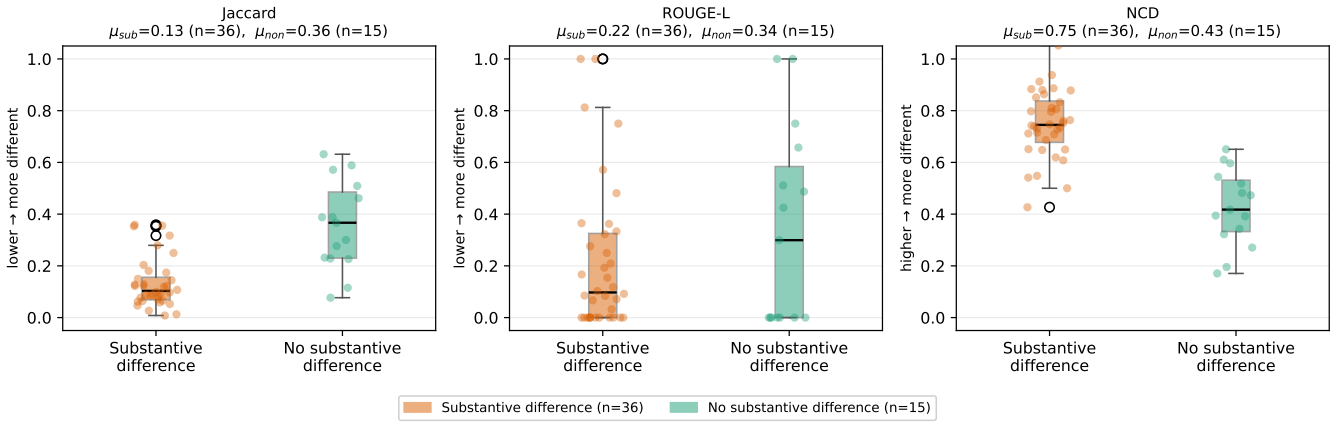

Figure 10: Similarity metrics split by whether an independent LLM judge classified the response pair as exhibiting a substantive semantic difference. NCD provides the clearest separation between the two groups.

separation (μ_{sub} = 0.75 vs. μ_{non} = 0.43), while Jaccard (0.13 vs. 0.36) and ROUGE-L (0.22 vs. 0.34) showed smaller gaps. Based on this, we adopted NCD with $\tau = 0.7$ as the operating point that best balanced probe sensitivity against participant disruption.

D Post-Condition Questionnaire

After each condition, we administered a post-condition survey using 7-point Likert scales (1 = strongly disagree, 7 = strongly agree). Table 1 lists all questionnaire items.

Dimension	Item
Task Awareness	"The system helped me maintain an overview of the overall context of my task."
Thought Structuring	"The system helped me organize and structure information and thoughts related to my task."
Perceived AI's Understanding	"The AI seemed to understand my task context and intentions during the task."
Authorship	"The system helped the task reflect my own thinking and judgment, not just the AI's suggestions."
Controllability	"The system gave me control over how my task context was organized and represented."
Overall Satisfaction	"I was satisfied with the overall interaction with the system."

Table 1: Post-condition questionnaire items.

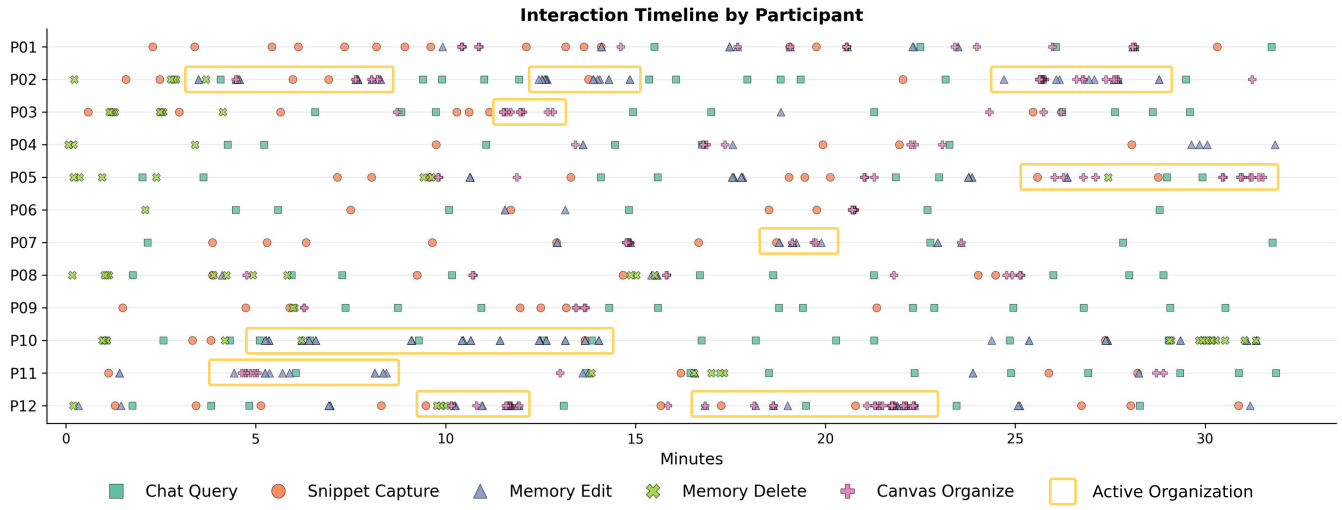

Figure 11: Interaction Log by Participant