IntentFlow: Investigating Fluid Dynamics of Intent Communication in Generative AI

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

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

Seoyoung Kim
School of Computing
KAIST
Daejeon, Republic of Korea
youthskim@kaist.ac.kr

Brandon Chin
College of Engineering
University of California Berkeley
Berkeley, California, USA
brandoncjw@hkn.eecs.berkeley.edu

Juho Kim
juhokim@kaist.ac.kr
School of Computing, KAIST
Daejeon, Republic of Korea
juho@skillbench.com
SkillBench
Santa Barbara, CA, USA

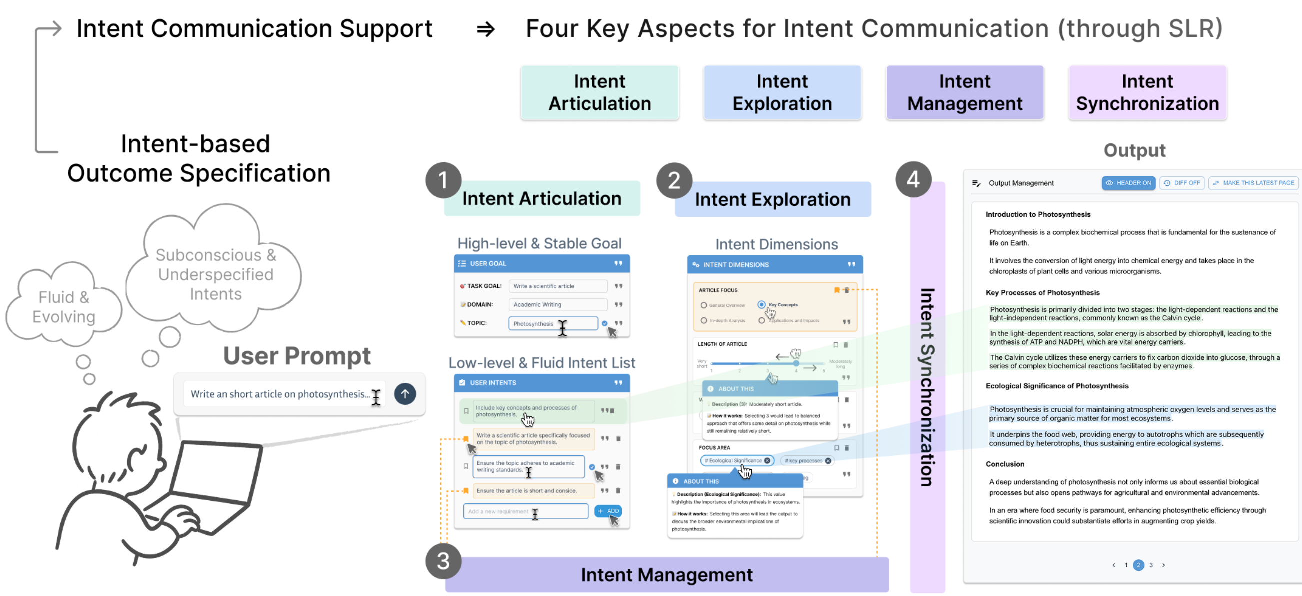

Figure 1: Our work addresses the challenges of intent-based outcome specification in human–LLM interaction, where user intents are often subconscious and underspecified, and tend to be fluid and evolving. Through a systematic literature review, we identified four aspects of intent communication support: • Articulation, • Exploration, • Management, and • Synchronization. We instantiate these aspects in IntentFlow, an LLM system for writing tasks that (1) helps users articulate vague or subconscious intents, (2) supports exploration of alternative or emerging directions, (3) manages evolving intents over time, and (4) synchronizes intents with generated outputs through linking.

ABSTRACT

Generative AI shifts interaction toward intent-based outcome specification, despite inherently vague, fluid, and evolving intents. While HCI research has proposed diverse interaction techniques to support this process, how key aspects of intent communication interplay to shape users' workflows remains underexplored. To bridge this gap, we conduct a systematic literature review of 46 HCI papers and identify four core aspects of intent communication support: intent • Articulation, • Exploration, • Management, and • Synchronization. To investigate how these aspects interplay in practice, we developed IntentFlow, a research probe that embodies all four aspects for a writing task, and conducted a comparative study (N=12). Our action-level behavioral analysis reveals that comprehensive

DIS '26, June 13–17, 2026, Singapore, Singapore

ACM ISBN 979-8-4007-2563-0/2026/06
https://doi.org/10.1145/3800645.3812999

support enables verification-driven refinement and progressive intent curation, reduces cognitive effort, and improves users' sense of control and understanding of intent–output alignment. We conclude with design implications for building generative AI systems that support intent communication as a dynamic, iterative process.

CCS CONCEPTS

• **Human-centered computing → Interactive systems and tools**; *Interaction paradigms*; *Empirical studies in HCI*.

KEYWORDS

Human-AI Interaction, Generative AI Systems, Intent Communication, AI alignment, Transparency, Writing Assistant

ACM Reference Format:
Yoonsu Kim, Kihoon Son, Seoyoung Kim, Brandon Chin, and Juho Kim. 2026. IntentFlow: Investigating Fluid Dynamics of Intent Communication in Generative AI. In *Designing Interactive Systems Conference (DIS '26), June 13–17, 2026, Singapore, Singapore.* ACM, New York, NY, USA, 34 pages. https://doi.org/10.1145/3800645.3812999

1 INTRODUCTION

Advances in generative AI, such as large language models (LLMs) and vision-language models (VLMs), allow people to easily obtain high-quality outputs, ranging from polished text [33, 39, 57, 82, 90] to functional code [45, 48] to complete visual designs [58], simply by describing what they want in a natural language prompt. This represents what Jakob Nielsen calls a new UI paradigm in computing history—*intent-based outcome specification*—where users tell the computer *what they want*, rather than *how to do it* [51].

While prompting may appear straightforward, communicating intent through natural language is inherently complex. Drawing from communication theory [24, 37, 67], we can understand prompting as conveying two key types of information: the user's **goal** and their **intent**. A **goal** denotes the high-level objective (e.g., writing a cover letter), which tends to be explicit and stable throughout the process. In contrast, **intents** capture low-level strategies, preferences, or constraints for achieving that goal, often emerging or shifting during the process (e.g., deciding whether to introduce one's background before the motivation, adopting a polite tone, or keeping certain sections concise). Communication scholars emphasize that intent is inherently fluid and often subconscious [24, 37, 38, 67]. People may not be fully aware of their own intents at the outset; these may evolve, surface gradually, or even fade as they reflect on outcomes and iterate on their actions.

HCI researchers have similarly noted that users' intents are often not fully articulated at the beginning of interaction [7, 22, 60, 71, 74], and a wide range of interaction techniques have been proposed to support intent communication over time [75, 78, 89]. However, these efforts are scattered across systems that address different aspects of intent communication in isolation, leaving open questions about how such support mechanisms interact and reinforce one another in practice.

To synthesize existing insights, we conducted a systematic literature review (SLR) of 46 HCI papers on generative AI systems related to intent communication. Through this review, we identified 14 interaction features and grouped them into four key aspects of intent communication: (1) **Intent Articulation**—helping users express vague intents, (2) **Intent Exploration**—supporting users in discovering and expanding their intents, (3) **Intent Management**—providing a structured way to manage evolving intents, and (4) **Intent Synchronization**—ensuring a mutual understanding between the user and LLMs regarding communicated intents (Table 1). While prior work has contributed valuable techniques for each aspect, how they interplay to shape users' dynamic intent communication process remains underexplored.

This gap motivates the following research questions, examining how the four aspects jointly shape intent communication in practice:

RQ1. How do intent communication support features affect users' perceived experience and cognitive workload?
RQ2. How do **users communicate their intents** when interacting with a system that integrates support for • Articulation, • Exploration, • Management, and • Synchronization?
RQ3. How does the interplay among different **intent communication support features** shape users' intent communication behaviors?

Addressing these questions requires a probe that integrates all four aspects within a single interaction workflow. We developed IntentFlow, a research probe that instantiates all four aspects—• Articulation, • Exploration, • Management, and • Synchronization—through representative and dominant features identified in our SLR. We situate IntentFlow in the context of LLM-based writing tasks, which provide a representative testbed due to their iterative nature and the need to balance multiple, evolving intents such as content structure, tone, and emphasis [19, 57].

Using this research probe, we conducted a within-subjects study (N=12) comparing IntentFlow with a conventional chat-based LLM interface representative of widely-used commercial systems such as ChatGPT Canvas [52] and Claude Artifact [1]. We include this baseline to contrast users' intent communication process when supported by all aspects versus the current practice, which primarily relies on conversational turn-taking with free-form prompts.

Our behavioral analysis revealed distinct interaction patterns when all four aspects of intent communication were jointly supported. Users engaged in iterative cycles of • Articulation and • Synchronization, using system feedback to verify how their intents were reflected in the output—demonstrating *verification-driven refinement*—and progressively consolidating tentative intents into stable configurations through *intent curation*. These patterns emerged from the interplay among aspects: • Synchronization served as a critical mediating mechanism, grounding • Exploration, verifying • Articulation, and informing • Management decisions; • Management provided stability that allowed • Articulation and • Exploration to build incrementally; and • Exploration revealed gaps that fed back into • Articulation. Users also reported significantly easier intent expression, improved understanding of intent–output alignment, and reduced cognitive effort.

Taken together, we derive four design implications suggesting that supporting intent communication as a dynamic process requires not just assembling known features, but designing how they interact across stages. Systems should (1) establish immediate bidirectional traceability between intents and outputs, (2) scaffold exploration within synchronized views to enable fluid iteration rather than restarting articulation, (3) support progressive commitment that enables comparison and refinement before committing to management, (4) surface existing intent configurations during articulation to expose conflicts, redundancies, and gaps in their current intent space.

In summary, this work makes the following contributions:

- A conceptualization of four key aspects that support intent communication—articulation, exploration, synchronization, and management—derived from a systematic literature review of 46 HCI papers on generative AI systems.
- Findings from a comparative user study demonstrating how comprehensive intent communication support reshapes user behavior and perception, reducing effort while improving control and intent–output alignment.
- Design implications for generative AI systems that support dynamic intent communication, emphasizing how features should interact across stages rather than operate in isolation.

2 RELATED WORK

2.1 Challenges of Aligning User Intent in Natural Language Interaction

Despite the remarkable capabilities of LLMs, several studies have consistently highlighted their limitations in understanding and aligning with user intent. Prior studies have emphasized that LLMs frequently generate responses that misalign with user expectations, sometimes leading to unintended consequences due to inherent biases or prompt sensitivity [2, 15, 28, 36, 79, 86]. One contributing factor lies in the interaction setup itself: natural language, the primary medium for prompting, provides important advantages by lowering the barrier to entry, enabling flexible self-expression without requiring specialized syntax [30, 86]. Yet the very qualities that make natural language accessible also introduce ambiguity and instability, making it difficult for users to articulate their intent with the precise phrasing for LLMs and forcing them into cognitively demanding trial-and-error reformulation [22, 34, 48, 50, 68, 86]. The opacity of the intent-to-output connection further worsens this issue: prompts may embed multiple intents, yet users are not informed which parts influenced the output, and small wording changes can yield unpredictable shifts [30, 50, 91]. Moreover, as user intent often evolves during interaction, current chat-based interfaces provide limited support for managing this fluid process. Intents expressed across multiple turns become fragmented within linear conversation histories, making it difficult for users to track, refine, and maintain alignment with model behavior [33, 85]. These challenges collectively highlight the need for new interaction mechanisms that help users articulate their intents, understand how their inputs influence the model, and manage them over time. Our work aims to address these issues by helping users more clearly articulate, refine, and adjust their intents within an LLM system.

2.2 Subconscious and Fluid Characteristics of User Intents

While intent alignment itself has been recognized as one of the biggest challenges for LLMs, this issue becomes even more complex when considering the characteristics of human intent during communication. Drawing from interpersonal communication research, we can distinguish user prompts into two types of information: **goals**, which are explicit and relatively stable, and **intents**, which refer to more fine-grained, sometimes subconscious strategies or actions that evolve dynamically throughout a conversation [37, 67]. This distinction is particularly relevant as users often interact with LLMs in ways that resemble human-to-human communication [86]. Research in cognitive task analysis reveals that intent is not spontaneously generated; rather, it emerges from a foundation of knowledge, cognitive processes, and goal configuration [14]. Consequently, user intents often shift during communication [37, 67], complicating how users interact with LLMs and making this a crucial consideration in interaction design. Furthermore, they can be particularly more apparent in creation tasks, such as writing or drawing, due to their iterative nature, requiring continuous reflection and refinement [7, 22, 60, 64, 72]. Recognizing intent as both subconscious and fluid thus highlights the need for interaction designs that can better accommodate the dynamic and situated nature of human communication with LLMs.

2.3 Supporting Intent-Aligned Interaction with Generative AI

There are diverse attempts to support users in interacting with generative AI systems in ways that better align with their intent. Prior work has explored supporting users to more effectively express their intent, such as providing prompt suggestions to scaffold ideation for text-to-image models [6], allowing easier prompting through direct manipulation [50]. Furthermore, to allow users to better attain the results aligning with their intents, there are attempts to help users understand LLMs' behaviors through a visual programming environment for hypothesis testing [3] or breaking down prompts into smaller subtasks and then aggregating the results [80].

For the cases where user intent is unclear or underspecified, there are attempts to go beyond passively responding to user prompts: proactively retrieving missing information when tasks are ambiguous [56] or asking for follow-up questions to better align with users' overarching goals [79]. For another approach to better understand user intents, IntentGPT identifies intent within user utterances, enhancing the system's ability to respond to varying goals with greater precision [59]. Moreover, to support changes in user intent, dynamic prompt middleware [17], which provides context-specific UI elements to better refine the user prompt, allows users to change their preferred options.

More recently, work has begun to reify intent as manipulable units to support more flexible and reflective workflows. IntentTagger introduces "intent tag"—atomic conceptual units that enable granular and non-linear micro-prompting, supporting intent elicitation and flexible workflows [22]. In a similar vein, AI Instruments embody prompts as reusable interface objects, reflecting multiple interpretations of ambiguous user intents and enabling iterative, non-linear exploration across creative tasks [58]. These approaches

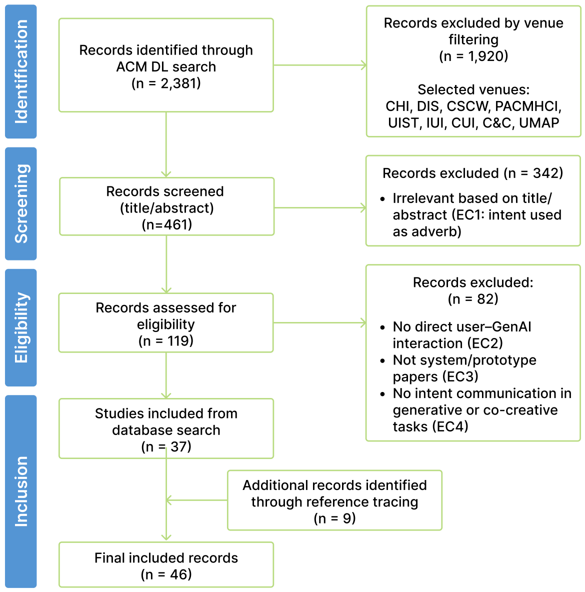

Figure 2: A PRISMA flow diagram illustrating the systematic literature review process across four stages: Identification, Screening, Eligibility, and Inclusion.

suggest new paradigms for structuring intent that go beyond linear prompting.

These efforts also exist in the context specific to writing with LLM as it involves cognitive content-generation that requires intensive and precise intent formulation and expression [23]. One line of research has proposed LLM-based writing-support systems that help users explore and prototype their writing while deciding on their writing intent [33, 69, 90]. There also exist approaches to help users better express their intent by allowing direct manipulation within the text to match their specific writing intent or style [50, 82]. Moreover, there are attempts to make AI-generated suggestions more explicit and manageable, for instance by structuring multiple variations side by side for rapid comparison or surfacing executable edits directly in the document with safeguards for accuracy [39, 57].

While prior work has improved how users express, reify, and align their intents with generative AI, it has rarely considered the subconscious and fluid nature of user intent as a basis for interaction design. We build on this body of work through a systematic literature review and present a system that enables more flexible and reflective interaction with users' evolving intents in writing tasks.

3 SYSTEMATIC LITERATURE REVIEW: INTENT COMMUNICATION SUPPORT

In this section, we present the process and results of our systematic literature review (SLR) of prior HCI research regarding intent communication support with generative AI systems. Our goal was to identify recurring *feature-level design patterns* for intent communication and to examine how these features have been combined—or separated—in existing systems.

3.1 Search and Filtering Process

We conducted our SLR following the PRISMA guideline [53], structuring the process into four stages: Identification, Screening, Eligibility, and Inclusion. Figure 2 presents the PRISMA flow diagram summarizing the number of records retained at each stage.

3.1.1 Identification. The ACM Digital Library was selected as our primary search source, as ACM venues represent the primary publication channels for interaction design research in human–AI systems. We searched for papers published between December 2022 and December 2025 using the following keyword combination in titles or abstracts: `("Generative AI" OR "LLM") AND ("intent" OR "intention")`. This search yielded 2,381 records. To focus on interaction design in human–AI systems, we filtered results to major ACM venues in HCI and closely related communities. Specifically, we included SIGCHI flagship and affiliated venues (e.g., CHI, DIS, CSCW, PACMHCI), along with closely related venues that contribute to human–AI interaction research, such as UIST, IUI, CUI, C&C, and UMAP. This scoping allowed us to capture a coherent yet diverse set of interaction-focused systems, while preserving a consistent lens on how interaction techniques support users' intent communication. After venue filtering, 461 papers remained. A complete list of selected venues is provided in Appendix A.1.

3.1.2 Screening. We conducted title and abstract screening on the 461 records. The first author performed this initial screening, as this stage focused on removing clearly irrelevant papers based on surface-level criteria rather than nuanced judgment. Specifically, we applied EC1 to exclude papers where the term "intent" did not refer to user intent communication (e.g., when used as an adverb such as "intentionally"), or where the work was clearly unrelated to intent communication in human–GenAI interaction. This step resulted in the exclusion of 342 papers, leaving 119 papers.

3.1.3 Eligibility. We assessed the full texts of the remaining 119 papers using three additional exclusion criteria (EC2–EC4) to filter papers that did not align with our SLR goal. This stage was conducted by two authors, who independently reviewed each paper and applied the criteria below:

- EC2: We focused on papers where users directly communicate their intent with GenAI or LLM systems. Studies where GenAI mediated intent communication between a user and a third party were excluded.
- EC3: We included only system or framework papers that proposed concrete interaction features, including prototypes. We excluded survey and workshop papers.
- EC4: We focused on papers involving co-creation or generative tasks (e.g., writing, coding, design) where users expressed their intent to the system and received outputs.

Disagreements were resolved through discussion to reach a consensus. Applying these criteria resulted in 82 exclusions, yielding 37 papers.

3.1.4 Inclusion. During the eligibility stage, we complemented the keyword-based search with reference tracing from the reviewed papers. This allowed us to identify additional candidate papers that were not retrieved in the initial search, particularly those that did not explicitly use the term "intent" but were relevant to intent communication based on their interaction design. All additional papers were assessed using the same inclusion criteria (EC1–EC4), and nine were included in the final dataset, resulting in a total of 46 papers. A complete list of included papers is provided in Appendix A.1.

3.2 Analysis

We conducted an open coding analysis to examine how prior systems support users' intent communication through interaction design. Three authors independently documented how each system supported users' intent communication, decomposing broader interaction patterns into atomic functional units where possible to minimize overlap. The authors then organized features through an affinity diagramming process, placing feature cards on a shared board and iteratively grouping them through discussion. While some features could plausibly belong to multiple aspects, they were assigned to the aspect that most centrally characterizes their primary function, determined through consensus. This process resulted in a codebook of 14 interaction features (Table 1), which capture recurring design patterns for supporting intent communication. To further illustrate how each feature has been instantiated in prior systems, we provide representative examples in Table 3. Building on this, we further abstracted these features into higher-level themes. Through this process, we identified four key aspects of intent communication. All coding and abstraction processes were conducted through iterative discussions among the three authors, and any disagreements during these processes were resolved through discussion until consensus was reached. Finally, we revisited all papers using the finalized coding scheme to ensure consistency.

3.3 Findings

3.3.1 Four Key Aspects of Supporting Intent Communication in Human-GenAI Interaction. Our analysis revealed four central aspects of how prior HCI systems supported intent communication:

- **Intent Articulation**: Helping users externalize their underspecified and vague intents into more concrete and actionable forms. This is a convergent process focused on transforming users' vague intent into specific instructions.
- **Intent Exploration**: Supporting users in discovering new, emerging intents that they may not have been initially aware of. This is a divergent process that encourages users to explore and expand their initial scope.
- **Intent Management**: Supporting users in organizing, revisiting, and curating their intents as they evolve over time. This is a stabilizing process focused on maintaining continuity across iterations by structuring intents into persistent and manageable representations.
- **Intent Synchronization**: Aligning the user's communicated intents with the LLM's output by making transparent how each intent is reflected in the generated output and how modifications of intents trigger corresponding updates. This process enables users to verify whether their intents are realized by the LLM as intended.

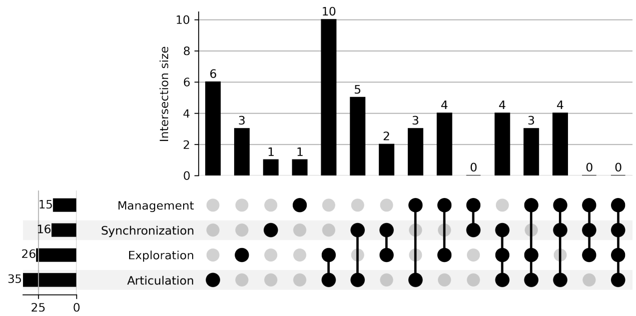

Figure 3: Weighted UpSet plot showing how existing systems combine four intent-support aspects—articulation, exploration, management, and synchronization. Each column represents a unique combination of supported aspects, and bar heights indicate the aggregated frequency of occurrences for each combination.

These aspects encompass the different yet interdependent ways in which systems can facilitate users in externalizing, expanding, managing, and aligning their intent during interactions with generative AI systems.

3.3.2 Aspect-level Coverage and Limited Integration Across Aspects. Prior work has predominantly focused on • Articulation and • Exploration, with many systems providing features that help users externalize intents more explicitly or explore alternative intents through variations, suggestions, or prompts (Table 1). In contrast, • Management and • Synchronization are less supported: relatively few systems offer explicit mechanisms for revisiting, curating, or relating multiple intents over time, or for helping users understand how their intents are interpreted and reflected in system outputs.

Examining how these aspects are combined reveals further scope for exploration. As shown in the weighted UpSet plot (Figure 3), • Articulation and • Exploration frequently co-occur, whereas combinations involving • Management and • Synchronization are comparatively rare, and no existing system supports all four aspects simultaneously. This coverage suggests that prior systems emphasize localized support for intent articulation or exploration, but do not provide integrated support for maintaining, coordinating, and aligning intents as interaction unfolds—often leaving users to manage these dynamics on their own.

3.3.3 Motivation for a Research Probe. While prior work has proposed a rich set of features within each aspect, there is a space to explore and understand how these aspects interact to shape users' intent communication behaviors during end-to-end interaction workflows by integrating four aspects. Prior work also points to challenges that arise when individual aspects are absent: without management, intents expressed across multiple turns risk becoming fragmented [85]; without synchronization, articulation can devolve into trial-and-error as users cannot verify how their intent is interpreted [48, 50]; without exploration, users may articulate within a narrow scope without discovering latent possibilities [69]. Exploring how all four aspects work together thus requires moving beyond retrospective analysis toward a probe system that deliberately embeds all four aspects in a unified interaction setting. In the

Theme	Feature	Paper	N	%
Articulation	**A1. Allowing users to specify and refine intent by directly referencing specific areas of the output.**	[13, 16, 31, 39, 44, 46, 47, 50, 68, 75, 77, 78, 87, 89, 90, 92]	16	34.78%
	A2. Decomposing users' vague input into granular sub-components	[9, 21, 26, 29, 31, 35, 39, 41, 42, 58, 69, 76, 88, 89]	14	30.43%
	A3. Supporting diverse modalities (e.g., sketch, image, metadata) for expressing user intent beyond text according to the task context	[13, 17, 22, 47, 54, 73, 83, 87, 92]	9	19.57%
	A4. Elaborating intent using users' vague inputs as seeds	[9, 32, 46, 69, 77, 84, 88]	7	15.22%
Exploration	**E1. Supporting navigation of intent variation spaces through output spectra**	[6, 8, 12, 22, 41, 42, 44, 49, 57, 69, 70, 87]	12	26.09%
	E2. Suggesting alternative intents	[11, 12, 21, 22, 39, 44, 46, 58, 61, 68]	11	23.91%
	E3. Supporting to remix intents or intermediate output	[31, 54, 65, 70, 76]	5	10.87%
	E4. Providing exploratory nudges through prompts and questions	[6, 21, 40, 68]	4	8.70%
Management	**M1. Structuring intents into manageable representations**	[6, 17, 22, 35, 55, 57, 58, 65, 78, 83, 90]	10	21.74%
	M2. Revisiting and curating past intents for independent editing or reuse	[12, 17, 70, 83, 84]	5	10.87%
	M3. Managing Relationships among Multiple Intents	[13, 75]	2	4.35%
Synchronization	**S1. Showing how intents are reflected in the output**	[9, 48, 54, 68, 83, 84]	6	13.04%
	S2. Previewing the effects of intent changes	[22, 44, 49, 50, 61, 75]	6	13.04%
	S3. Exposing the system's intent interpretation	[9, 35, 47, 48, 73, 92]	6	13.04%

Table 1: Interaction features for intent communication identified through our SLR, organized into four themes. For each feature, we report the number and percentage of reviewed papers that incorporated it. Features highlighted in bold indicate the representative interaction features selected to instantiate the design of our research probe (IntentFlow), based on their prevalence and centrality in prior systems.

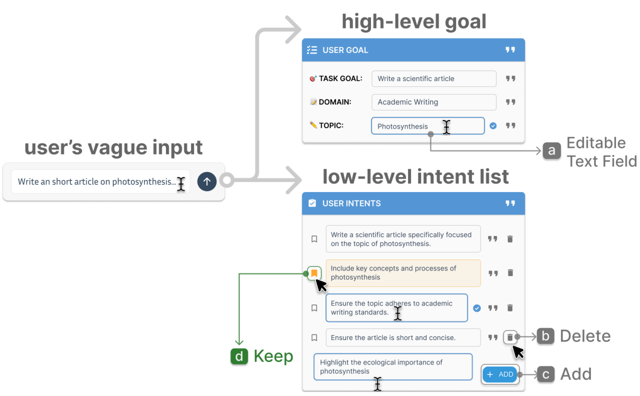

Figure 4: Articulation support in IntentFlow. The system decomposes a user's vague input into a high-level goal and a set of low-level intents, externalized as editable components. Users can (a) edit, (b) delete, or (c) add intents to refine their intent articulation. (d) Intents can also be pinned to support intent management across iterations.

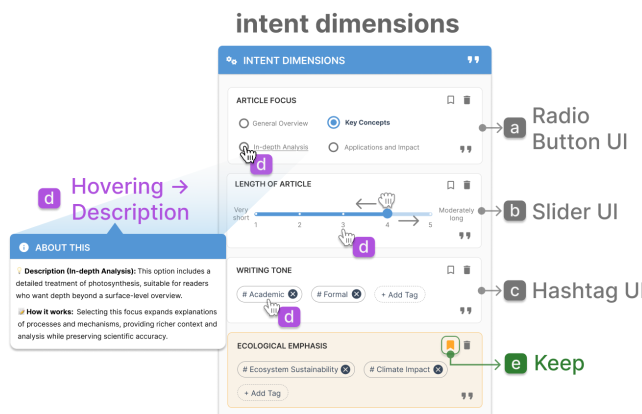

Figure 5: Exploration support in IntentFlow. Intent dimensions are presented through multiple UI controls, including (a) radio buttons, (b) sliders, and (c) tags, enabling users to explore alternative intent configurations. The figure also shows (d) a preview mechanism for synchronization and (e) a pinning option for intent management.

following section, we describe how representative features were selected for each aspect and how they are implemented in practice.

4 RESEARCH PROBE: INTENTFLOW

We designed a system, IntentFlow, as a research probe [5] that integrates support for articulation, exploration, management, and synchronization within a single interaction workflow, enabling these aspects to interplay during intent communication. The goal is not to evaluate individual features in isolation, but to examine how the four aspects interact and jointly shape users' intent communication behaviors when they operate together—generating theoretically grounded insights that can inform future system design. We acknowledge that this design does not allow us to isolate the contribution of individual aspects; we address this as a limitation in Section 7.4.

Grounded in representative features from prior work identified through our SLR (Table 1), IntentFlow brings together these aspects in a unified setting. In this section, we describe the design rationale behind IntentFlow, focusing on how features were selected for each aspect and how they are realized through the system's interface and underlying pipeline.

4.1 Design Rationale and Feature Selection

We selected dominant and representative features for each intent-support aspect to instantiate the probe, which are highlighted in bold in Table 1. Our goal was not to exhaustively implement all prior features, but to capture the core and dominant interaction affordances that characterize each aspect.

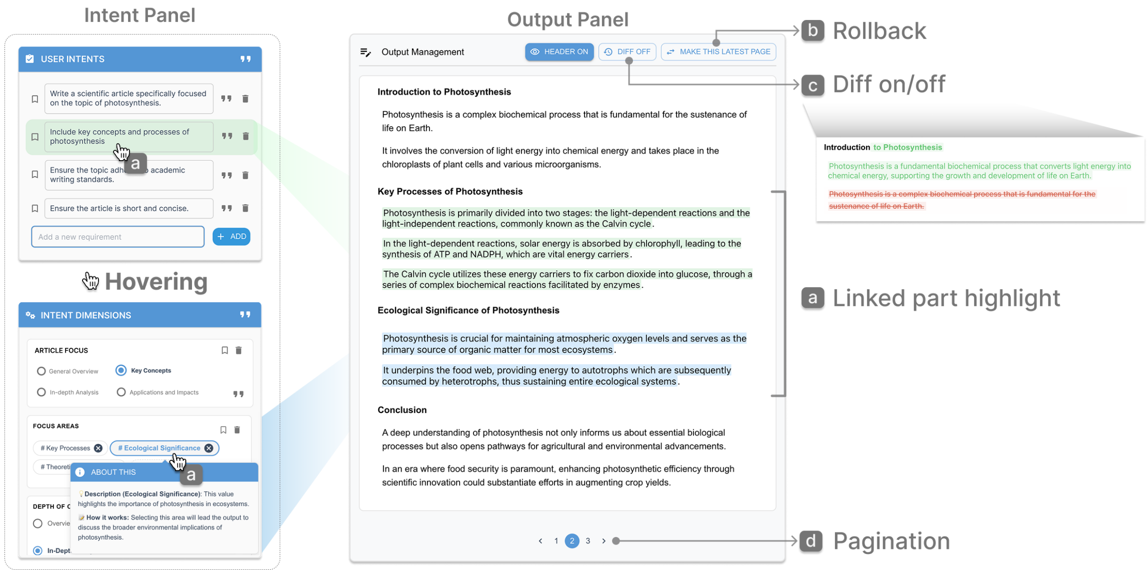

Figure 6: Synchronization support in IntentFlow. (a) Hovering over intents and intent dimension values highlights corresponding linked parts of the generated output in green (intents) or blue (intent dimension values). (b) Users can roll back to any prior version, which brings the selected output and its associated intents to the latest position in the workflow. (c) Diff view compares old and new outputs. (d) Pagination allows users to navigate and manage multiple output versions over time, supporting intent management.

For • Articulation, prior work most commonly supported it by decomposing users' vague input into granular sub-components and enabling direct manipulation to individual intents (A1, A2 in Table 1). Reflecting this, IntentFlow decomposes a user's chat prompt into a structured representation consisting of a *high-level goal* and a set of *low-level intents*, which are externalized as editable components (Figure 4). Users can directly revise, add, or remove individual intents, enabling them to progressively articulate and refine their intent beyond an initial vague prompt.

To support • Exploration, we drew on prior systems that enabled users to navigate intent variations and consider alternatives through output spectra or adjustable parameters (E1, E2 in Table 1). In IntentFlow, exploration is supported through the *intent dimension section*, which exposes adjustable dimensions for each intent using multiple UI controls such as radio buttons, sliders, and tags (Figure 5). Users can explore how different emphases or alternative options would shape the resulting output by directly manipulating these dimensions.

For • Management, our SLR highlighted features that structure intents into manageable representations and support revisiting past intents over time (M1, M2 in Table 1). To instantiate these supports, IntentFlow treats intents and their associated dimensions as persistent entities that users can pin or unpin across interaction turns, directly edit, and revisit through versioning (Figure 4d, Figure 5e, Figure 6d). This design allows users to curate an evolving set of intents, supporting longer-term intent communication beyond early-stage ideation and enabling us to study how users manage multiple intents across iterative workflows.

Finally, for • Synchronization, prior work has emphasized helping users preview the effects of intent changes and see how their intents are reflected in the generated output (S1, S2 in Table 1). In IntentFlow, synchronization is supported by explicitly linking user intents and intent dimension choices to corresponding segments in the generated output (Figure 6) and providing contextual previews for alternative dimension values (Figure 5d). This design allows users to verify how their intents are reflected in the output and to iteratively adjust this alignment as interaction unfolds.

4.2 IntentFlow Interface

We instantiate IntentFlow in the context of LLM-based writing tasks, which provide a suitable testbed for studying intent communication due to their iterative, multi-faceted, and evolving nature [19, 57]. Writing tasks often require users to articulate high-level goals, refine multiple interacting intents, explore alternative framings, and iteratively align system outputs with evolving expectations, making them well-suited for examining how intent-support mechanisms operate over time.

As shown in Figure 7, the interface of IntentFlow is organized into three main panels: a **Chat Panel**, an **Intent Panel**, and an **Output Panel**. Together, these panels support a continuous workflow in which users can express intents, refine and manage them, and inspect how they are reflected in the generated output.

The **Chat Panel** serves as the primary entry point for user input. Users provide an initial, often underspecified prompt describing their writing task. Rather than treating this input as a single-shot instruction, IntentFlow uses it as a starting point for structuring intent communication, triggering the extraction of a high-level goal, low-level intents, and intent dimensions that populate the **Intent Panel**.

The **Intent Panel** externalizes users' intents into structured, editable representations and consists of three sections. The *Goal Section* captures stable, high-level aspects of the task, such as the

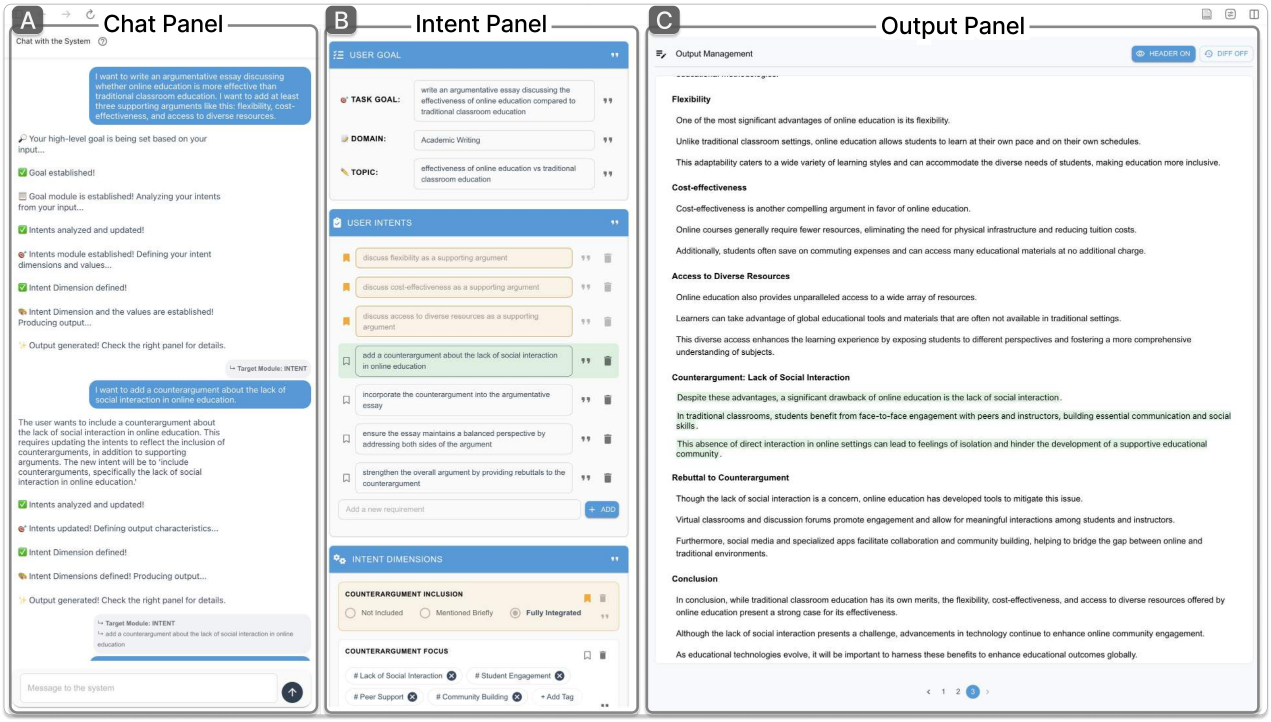

Figure 7: Overall interface of IntentFlow. It is split up into three panels: (A) Chat Panel, (B) Intent Panel, and (C) Output Panel.

overall writing goal, domain, and topic. Making these elements explicit allows users to verify and revise the system's understanding of their overarching objective. The *Intent List Section* displays a set of low-level, discrete intents inferred from the user's input. In addition to explicitly stated intents, the system also surfaces implicit intents that are logically required to carry out the task, reflecting how the LLM decomposes an underspecified prompt into fine-grained subtasks. Each intent is presented as an editable item that users can revise, add, or delete, supporting progressive • Articulation as users clarify and refine what they want to achieve. The *Intent Dimension Section* further decomposes each intent into adjustable dimensions, exposing parameters such as emphasis, scope, or style through interactive controls. The UI format and initial value of each dimension (e.g., radio buttons, sliders, or tags) are automatically selected based on the characteristics of the associated intent. By directly manipulating these dimensions, users can explore alternative interpretations and configurations of their intents, supporting • Exploration during the writing process. In addition, hover-based explanations reveal how different dimension values would affect the generated output, supporting • Synchronization by helping users anticipate how intent refinements will be reflected in the system's behavior. Across both sections, users can pin ( ) selected intents and dimension values to persist them across interaction turns, supporting • Management by allowing them to manage which aspects of their evolving intent should be retained as the writing process unfolds.

The **Output Panel** displays the text generated by the LLM based on the current configuration of goals, intents, and intent dimensions. Beyond presenting the output, this panel primarily supports • Synchronization by making the relationship between user intent and generated text explicit. Hovering over an intent or a dimension value highlights the corresponding segments in the output, enabling users to verify how specific intent choices are reflected in the generated content. This panel also supports • Management by maintaining a version history of each output along with its corresponding intent configuration from the **Intent Panel**. Users can revisit prior intent-output states, compare changes using a diff view, and roll back to a previous state to continue their work from an earlier configuration. Together, these features allow users to manage, revisit, and refine their intent configurations over time while maintaining alignment between intent and output.

4.3 Internal Pipeline and Implementation

Internally, IntentFlow operates through a modular LLM pipeline that incrementally translates user input and interaction into structured representations of goals, intents, and intent dimensions, which in turn condition output generation. As illustrated in Figure 8, user inputs and edits are processed through dedicated modules that extract and update intent representations, generate text based on the current intent state, and establish links between intents, dimension values, and corresponding output segments. The pipeline is designed to preserve and update intent-related state, enabling • Articulation, • Exploration, • Management, and • Synchronization to be sustained over time.

We implemented IntentFlow as a web-based application with React for the frontend and Flask for the backend. The backend leveraged the OpenAI API to handle language model functionalities, with each module responsible for a specific role in the pipeline,

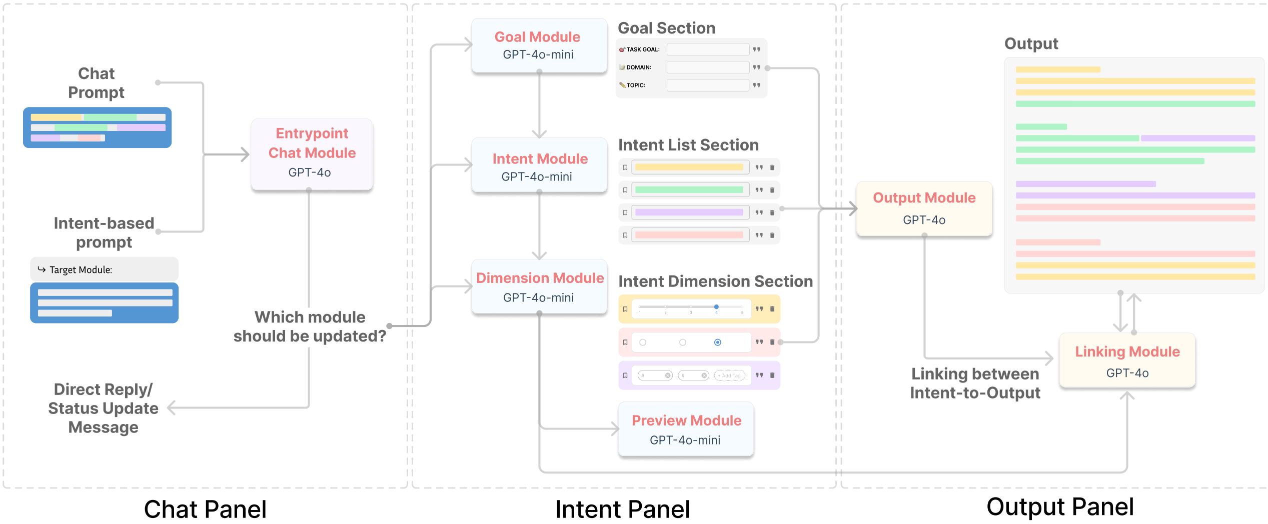

Figure 8: Internal pipeline of IntentFlow. The *EntryPoint Chat Module* first interprets the prompt and coordinates three intent-processing modules: the *Goal Module*, *Intent Module*, and *Dimension Module*. These modules extract and structure the user's goal and intents, which are then used by the *Output Module* to generate text. The *Preview Module* provides brief explanations of dimension values and their potential effects on the output. The *Linking Module* associates each intent with corresponding segments in the generated output. Each module box shows its LLM model name.

as illustrated in Figure 8. To improve responsiveness in real-time interactions, we used a smaller model for the *Goal*, *Intent*, and *Dimension* modules, and further optimized the *Linking Module* using multiprocessing to reduce latency. Full prompts for each module are provided in Appendix A.2 .

4.4 Pipeline Validation

Before conducting the user study, we performed a lightweight validation to verify that IntentFlow's intent-processing pipeline (Figure 8) operates reliably across diverse writing contexts. Specifically, we examined whether the pipeline could (1) extract coherent task goals and intents from underspecified prompts, (2) generate relevant and interpretable intent dimensions with appropriate UI representations, and (3) correctly associate intents and dimension values with corresponding parts of the generated output.

4.4.1 Setup. We selected 12 representative prompts—two from each of the six writing contexts defined by Lee et al. [43]: academic, creative, journalistic, personal, professional, and technical. For each prompt, we generated a full pipeline output and created a corresponding survey form for evaluation. To reduce ambiguity and individual variance, the survey used binary (Yes/No) questions, complemented by an optional free-form section. We recruited 60 evaluators (5 per prompt) via Prolific [1]. To ensure evaluation quality, we excluded evaluators who had never used LLMs, lacked experience with the given writing task, or reported unfamiliarity with the given writing topic. Each evaluator was compensated with £5 for completing a 30-minute evaluation task. Detailed information about the prompts, writing tasks, and topics used in the evaluation is provided in the Appendix A.3.2.

[1]https://prolific.com

Evaluation Criteria	"Agree"(%)
Q1. Goal Alignment	95.00
Q2. Set of Intents – Completeness	95.00
Q3. Set of Intents – Distinctiveness	86.67
Q4. Individual Intents – Relevance	94.08
Q5. Intent Dimension – Relevance	86.56
Q6. Intent Dimension – UI Appropriateness	86.78
Q7. Intent Dimension – Value Appropriateness	86.14
Q8. Intent-to-Output Linking – Link Accuracy	94.04

Table 2: Evaluation results for each question in the technical evaluation. Values indicate the percentage of "Agree" responses aggregated across all prompts and participants.

4.4.2 Validation Criteria. We designed questions targeting each pipeline module to assess whether the extracted goal reflected the user's overall objective (Q1. Goal Alignment); the set of intents was reasonably complete (Q2), distinct (Q3), and relevant (Q4); intent dimensions were relevant (Q5) and paired with appropriate UI controls and values (Q6–Q7); and highlighted output segments accurately reflected the corresponding intents (Q8. Link Accuracy). Full question wording is provided in Appendix A.3.1.

4.4.3 Results. As summarized in Table 2, over 85% of responses were positive across all criteria, indicating that the pipeline generally produces coherent and interpretable intent representations and links. We also analyzed qualitative feedback from evaluators. Several participants noted that when intent dimensions were presented without clear descriptions of value meanings, it was difficult to judge their appropriateness. For example, when the "Formality Level" dimension was presented as a slider with an initial value of 4, participants reported that it was hard to judge how formal the value 4 actually was. This likely contributed to the lower agreement in the *Intent Dimension* category since our technical evaluation did not provide a hover-based explanation for each dimension value. This underscores the importance of providing descriptive explanations for dimension values (Figure 5d).

5 USER STUDY

To understand how the four intent communication supports shape users' behaviors and experiences compared to current practice, we conducted a within-subjects user study (N=12) comparing IntentFlow with a conventional chat-based LLM interface representative of widely used commercial systems. Rather than isolating the effect of individual features, our goal is to use IntentFlow as a ***research probe*** to observe how different forms of intent support are taken up by users in practice. The research questions guiding this investigation are introduced in Section 1. In particular, we centered our analysis on **users' action-level intent communication behaviors**—such as adding, adjusting, correcting, and removing intents—captured through participants' post-hoc annotations of their own inputs. We analyzed these behaviors alongside system logs that recorded **participants' use of different intent communication support features** (e.g., articulation, exploration, management, and synchronization), as well as **self-report surveys** and **post-study interviews** to capture users' subjective experiences and interpretations. To ground this investigation in users' existing practices, we implemented a baseline system that reflects common chat-based LLM writing interfaces such as ChatGPT Canvas [52] and Claude Artifact [1]. The baseline represented an ecologically valid status quo for LLM-based writing, in which users primarily communicate intent through linear, conversational turn-taking with free-form prompts. The baseline used the same generation model as IntentFlow, `gpt-4o-2024-08-06`. The screenshot of the baseline interface can be found in Figure 17 in the Appendix.

5.1 Participants

We recruited 12 participants (8 male, 4 female; age M = 25.50, age SD = 2.75) through online recruitment posting at our university community platforms. Participants were not professional writers; however, all had prior experience with writing tasks and using large language models (LLMs). We intentionally recruited participants with writing and LLM experience to observe how experienced writers manage complex, nuanced, and evolving intents during interaction. During recruitment, we administered a pre-survey on participants' LLM experience, writing background, and topic familiarity, excluding those with insufficient knowledge or experience. Participants received 30,000 KRW (around 22 USD) for the 90-minute session, and to encourage careful and motivated engagement, we offered an additional 20,000 KRW (14 USD) performance-based bonus to the top 40% of participants. Additional details, including participants' LLM experience and writing experiences, are provided in the Appendix A.4.4.

5.2 Tasks

Each participant completed two writing tasks, which were randomly paired with the counterbalanced study conditions. We selected writing tasks commonly used in prior HCI studies [20, 33, 57] and added short scenarios to each task to introduce more nuanced and multifaceted considerations. The first task involved **writing a social media post** explaining a scientific concept (e.g., the Doppler Effect) for a general audience with little scientific background. The second involved **writing a professional email** applying for a personal secretary position for a well-known individual outside the participant's domain. Through multiple rounds of pilot testing, we refined the scenarios to ensure that the two tasks were similar in complexity and involved a comparable level of multifaceted consideration on writing intents—such as tone, structure, and content depth. To guide the output scope, we set a flexible length guideline of around half an A4 page. We intentionally did not specify an exact word count to prevent users from fixating on meeting a numerical target, which could distract from focusing on the given task scenario and goal. Importantly, as our study centered on intent communication, participants were encouraged to explore and refine intents rather than produce polished drafts. Full task descriptions are available in Appendix A.4.1.

5.3 Procedure

The study was conducted either in person or online via Zoom [2], depending on participant availability. Each study session lasted approximately 90 minutes and followed a fixed protocol (Figure 9). After a brief introduction and consent process (5 minutes), participants were given a tutorial for the first system (8 minutes). They then completed the first writing task using that system (20 minutes), followed by a post-survey and an annotation activity (10 minutes). To gather each participant's intent communication actions, we brought them back to the system after the survey to review their interactions and annotate the purpose of each input. These annotations were used in our analysis of how users engaged in intent communication, as described in more detail in Section 5.4. This process was then repeated for the second system and task. At the end of the sessions, participants completed a semi-structured interview (10 minutes) reflecting on their experience with both systems, including their preferences, their strategies for expressing and adjusting intent, and the usefulness and challenges of each system.

5.4 Measures

To address our research questions, we collected and analyzed three complementary data sources: (1) annotated intent communication actions derived from user inputs, (2) system logs capturing participants' use of intent communication support features, and (3) self-report surveys and post-study interviews.

5.4.1 ***Intent Communication Action Annotation***. To examine how users enacted intent communication over time (RQ2), we conducted a fine-grained analysis of participants' intent communication actions throughout the task process. We categorized all user inputs into four types of intent communication actions: **Add**, **Delete**, **Correct**, or **Adjust**. This taxonomy was informed by prior work examining how users express, refine, and repair intent during interactions with LLM-based systems [4, 10, 36, 63]. **Add** refers to introducing a new intent not previously expressed. **Delete** refers to removing a previously expressed intent; for instance, a user may

[2]https://zoom.us/

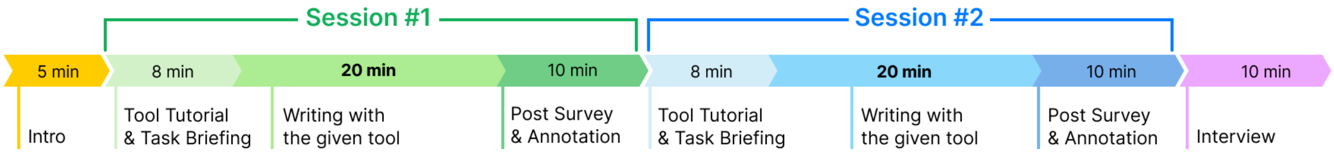

Figure 9: User study procedure. After a brief 5-minute introduction, each participant completed two sessions. The first 8 minutes were for the system tutorial and task briefing. The user had 20 minutes to write using the given system, followed by a 10-minute post-survey and annotation. At the end of both sessions, there was a 10-minute interview.

have initially requested a summary at the end but later decided to omit it. **Correct** captures cases where a user re-communicates an earlier intent due to a misunderstanding or misalignment by the system; for example, the user asks for a concise style, but the system generates something too verbose, prompting the user to restate the intent more clearly. Finally, **Adjust** involves modifying an existing intent in terms of degree or nuance, for example, slightly increasing the level of formality or adjusting the specificity of a detail without changing the underlying intent altogether.

After completing each task and post-survey, participants were brought back to the system and asked to review their interaction history and annotate each of their inputs with an intent communication action. For each input, they indicated which action type best reflected their intent at the time of interaction. For interface-level interactions in IntentFlow that unambiguously corresponded to a specific action type—such as directly adding or deleting intent items, or adjusting sliders and radio buttons—we automatically labeled the corresponding action and did not require manual annotation. This post-hoc annotation process allowed us to capture participants' *intended communicative function* behind each input, enabling fine-grained analysis of how intent communication evolved over the course of the writing process.

*5.4.2 **System Logs of Intent Communication Support Feature Usage**.* To examine how *intent communication support features* mediated users' behaviors (RQ3), we analyzed system logs recording when and how participants engaged with features supporting • Articulation, • Exploration, • Management, and • Synchronization. By aligning these logs with annotated intent communication actions, we examined how specific features shaped users' intent communication behaviors over time.

*5.4.3 **Self-Report Ratings**.* To capture participants' subjective experiences with each system (RQ1), we collected self-report ratings after each task using 7-point Likert scales (1: Strongly Disagree, 7: Strongly Agree). The survey items were adapted from prior work on human–AI interaction, particularly studies on LLM-supported writing systems [33, 57, 80]. They measured participants' perceived *ease* and *clarity* of intent expression, intent *discovery* and *elaboration*, *transparency* and *understanding* of intent–output relationships, and *alignment* between users' intents and generated outputs. The full list of survey items (M1–M11) is provided in the Appendix A.4.2. In addition, participants completed the NASA Task Load Index (NASA-TLX) [25] using a 7-point Likert scale to assess perceived workload.

5.4.4 Post-study Interviews. After completing both sessions, participants completed a semi-structured interview to reflect on their experiences with both systems. We asked how they approached expressing, revising, and discarding intents during the task process, how their strategies may have differed across systems, and how they perceived the role of explicit intent representations in supporting or constraining their workflow. In addition, participants were asked to reflect on whether and how the final set of intents they created or kept during the task might be reused in future writing tasks. The interview data were used to complement our quantitative analysis by providing contextual insights into interaction patterns and feature usage, as well as participants' mental models of intent communication workflows.

6 RESULT

For all statistical comparisons, we first conducted Shapiro-Wilk tests to examine the normality of each measure. Based on the results, we used paired t-tests for those that met the normality assumption ($p \geq .05$) and Wilcoxon signed-rank tests otherwise.

6.1 RQ1. How do intent communication support features affect users' subjective experience?

To examine how intent communication support features affect users' subjective experiences, we analyzed participants' self-report ratings (M1–M11) and perceived workload using NASA-TLX.

6.1.1 Perceived support for intent expression, alignment, and control. Across all intent communication measures, participants rated IntentFlow significantly higher than the Baseline (M1–M11; all $p < .05$; Figure 10). In particular, participants reported substantially greater ease and clarity of intent expression (M1–M2), stronger support for discovering and elaborating intents (M3, M8), and improved understanding of how their intents were reflected in the generated output (M4–M6). Ratings related to intent–output alignment and adjustment were also consistently higher in IntentFlow (M7, M9), indicating that participants felt better able to steer the system toward their intended direction. Ratings for perceived control and reusability (M10–M11) further suggest that participants viewed the intent communication process as a transferable strategy beyond the current task.

6.1.2 Reduced cognitive workload and frustration. NASA-TLX results further show that these perceived benefits were accompanied by reduced cognitive effort. Participants reported significantly lower overall workload when using IntentFlow compared to the Baseline (IntentFlow: $M = 15.67$, $SD = 4.01$; Baseline: $M = 19.67$, $SD = 4.50$; $p = .004$, $t = 2.02$). A breakdown of individual NASA-TLX dimensions revealed that this reduction was driven primarily

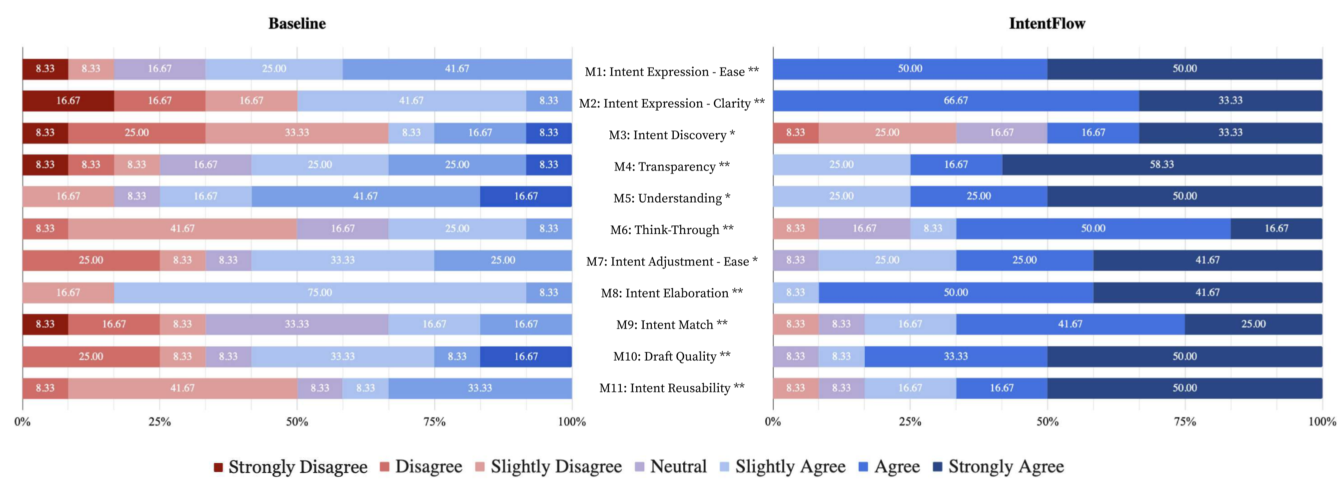

Figure 10: Distribution of participants' ratings on intent communication experience. (*$p < .05$, **$p < .01$)

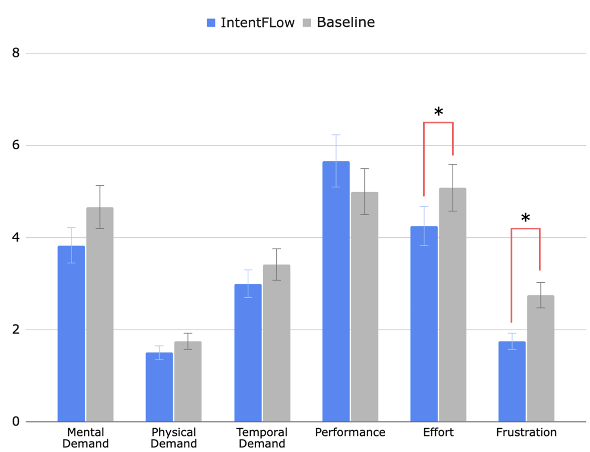

Figure 11: Comparison of NASA-TLX ratings between IntentFlow and Baseline. (*$p < .05$).

by lower perceived **Effort** (IntentFlow: $M = 4.25$, $SD = 1.22$; Baseline: $M = 5.08$, $SD = 0.90$; $p = .048$) and **Frustration** (IntentFlow: $M = 2.75$, $SD = 1.49$; Baseline: $M = 3.83$, $SD = 1.75$; $p = .034$), while differences in mental, temporal, and physical demand were less pronounced (Figure 11). These findings suggest that IntentFlow did not reduce workload by simplifying the task itself, but by reducing the cognitive overhead associated with maintaining, restating, and repairing intent.

6.2 RQ2. How do users communicate their intents with IntentFlow?

To examine how intent communication unfolds over time, we analyzed participants' annotated interaction logs across four intent communication action types—**Add**, **Delete**, **Correct**, and **Adjust**—comparing their distributions, temporal sequences, and action-to-action transitions between the Baseline and IntentFlow.

6.2.1 Overall differences in intent communication actions. As shown in Figure 13, participants' intent communication behaviors differed substantially across the two systems. Participants performed significantly fewer **Correct** actions when using IntentFlow than in the Baseline (Action Count: IntentFlow: $M = 0.50$, $SD = 0.67$; Baseline: $M = 4.33$, $SD = 2.64$; $p < .001$, $t = -4.81$), indicating fewer instances where users had to restate previously expressed intents due to system misalignment. In contrast, the number of **Add** actions was comparable across systems (IntentFlow: $M = 3.75$, $SD = 2.05$; Baseline: $M = 4.42$, $SD = 1.38$; $p = 0.296$, $t = -1.10$), suggesting that users introduced new intents at a similar rate regardless of interface. However, IntentFlow prompted significantly more **Adjust** actions (IntentFlow: $M = 4.50$, $SD = 2.97$; Baseline: $M = 1.17$, $SD = 0.72$; $p = .005$, $t = 3.46$) and more frequent **Delete** actions (IntentFlow: $M = 1.00$, $SD = 1.13$; Baseline: $M = 0.17$, $SD = 0.39$; $p = .031$, $W = 21.00$). These differences indicate a shift from repeatedly correcting misunderstood intents toward actively refining and curating existing intents.

6.2.2 Temporal evolution of intent communication. Beyond aggregate counts, the temporal distribution of actions further illustrates how intent communication evolved during the writing process (Figure 12). In the Baseline, early stages exhibited diverse transitions such as **Add**→**Add**, **Add**→**Adjust**, and **Add**→**Correct**. Over time, however, interactions increasingly converged on **Correct**-centered patterns, including repeated **Correct**→**Correct** transitions and frequent alternation between **Correct** and • **Rollback**. This pattern reflects users repeatedly attempting to reassert previously stated intents or reverting outputs after misalignment, rather than incrementally refining them. In contrast, IntentFlow showed a different trajectory. Early interactions similarly involved exploratory patterns such as **Add**→**Adjust** and **Add**→**Delete**, indicating users externalizing and probing potential intents. As the session progressed, however, intent communication increasingly centered on **Adjust**, with users repeatedly revisiting and fine-tuning existing intents rather than restating them. This temporal shift suggests that

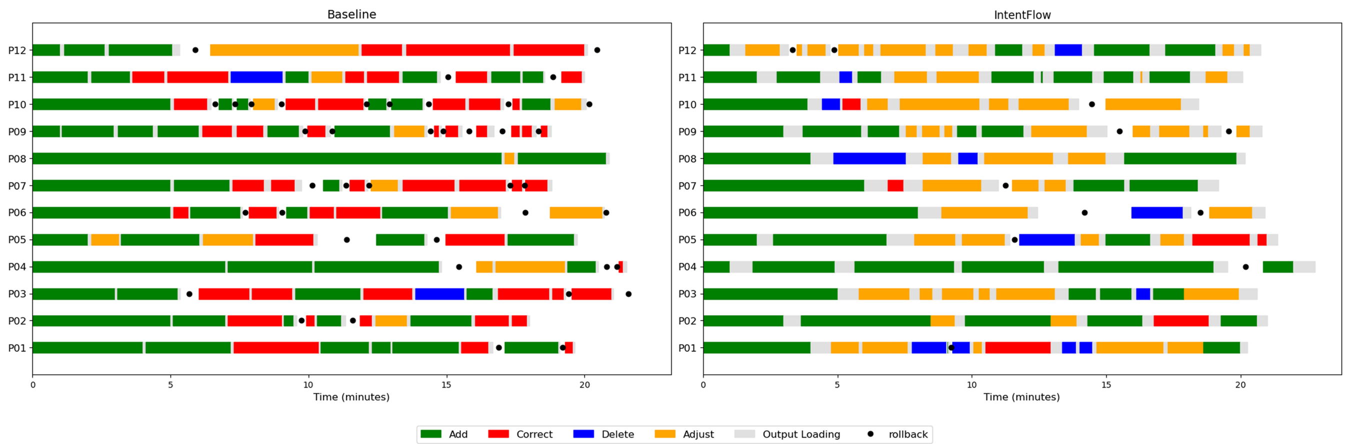

Figure 12: Action log sequences for 12 participants in each condition: Baseline and IntentFlow.

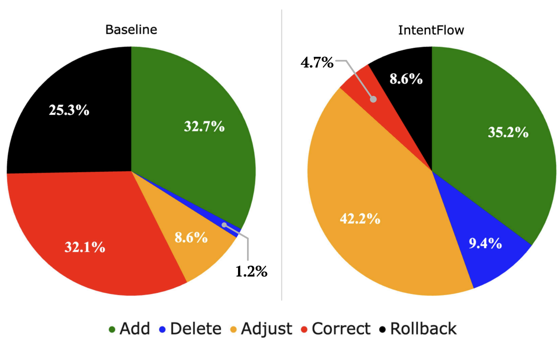

Figure 13: Percentage distribution of action types aggregated across all participants. Each chart reflects the proportion of actions by type in IntentFlow and Baseline.

intent communication in IntentFlow stabilized around refinement rather than correction.

6.2.3 Transition structure and convergence patterns. To further characterize these differences, we analyzed normalized action transition matrices for each system (Figure 14). In the Baseline, transition probabilities were concentrated around **Correct** and • **Rollback** indicating recurrent breakdown-recovery cycles. By contrast, IntentFlow exhibited a transition structure centered on **Adjust** action. **Adjust**→**Adjust** emerged as the most dominant transition, and transitions to **Correct** or • **Rollback** rarely appeared. The difference matrix further highlights this contrast: compared to the Baseline, IntentFlow showed substantially higher transition probabilities into **Adjust** and lower probabilities into **Correct**. These results quantitatively confirm that intent communication in IntentFlow converged toward adjustment-centered refinement, whereas the Baseline tended to funnel users into correction- and rollback-driven cycles.

6.2.4 Divergent roles of rollback across systems. We observed that users' use of • **Rollback** differed between the two systems. Rollback occurred significantly more often in the Baseline than in IntentFlow (Action Count: IntentFlow: $M = 0.92$, $SD = 0.79$; Baseline: $M = 3.42$, $SD = 2.50$; $p = .003$, $t = -3.80$), but with qualitatively different usage patterns. In the Baseline, • **Rollback** was most often paired with **Correct** actions and functioned as a *reset mechanism*: users reverted to earlier outputs after the system failed to respect previously stated intents, then restated or reissued similar instructions. Two participants (P9, P10) explicitly noted frustration, as P9 noted, *"In the B (Baseline), I asked to change only a specific part, but the whole output changed, so I often had to rollback and reissue the same prompt."* In IntentFlow, • **Rollback** more frequently appeared alongside **Adjust** actions, supporting controlled comparison and refinement. Four participants (P7, P9–10, P12) described experimenting with different intent adjustments, comparing resulting outputs, and rolling back to a version that better aligned with their intent. For example, P10 explained, *"In the A (IntentFlow), having the intent dimensions surfaced in the UI made it easier to explore and adjust. I tried different variations and rolled back to the output I liked most."* This indicates that • **Rollback** in IntentFlow supported *exploratory refinement* rather than recovery from breakdowns.

6.2.5 Dominant transition patterns within IntentFlow. To further characterize how intent communication unfolded within IntentFlow, we analyzed the distribution of action transitions observed in this condition (Figure 15). The most frequent transition was **Adjust**→**Adjust**, accounting for 24.5% of all transitions, followed by **Add**→**Adjust** (16.3%), **Add**→**Add** (16.3%), and **Adjust**→**Add** (11.2%). Together, these four transitions accounted for 68.4% of all transitions, indicating that a small set of transition patterns dominated users' intent communication in IntentFlow. These patterns reflect a process centered on iterative refinement and structured exploration, rather than repeated correction of misaligned outputs.

6.3 RQ3. How do different intent communication support features mediate the intent communication process?

To understand how different intent communication support features shaped users' intent communication behaviors, we examined

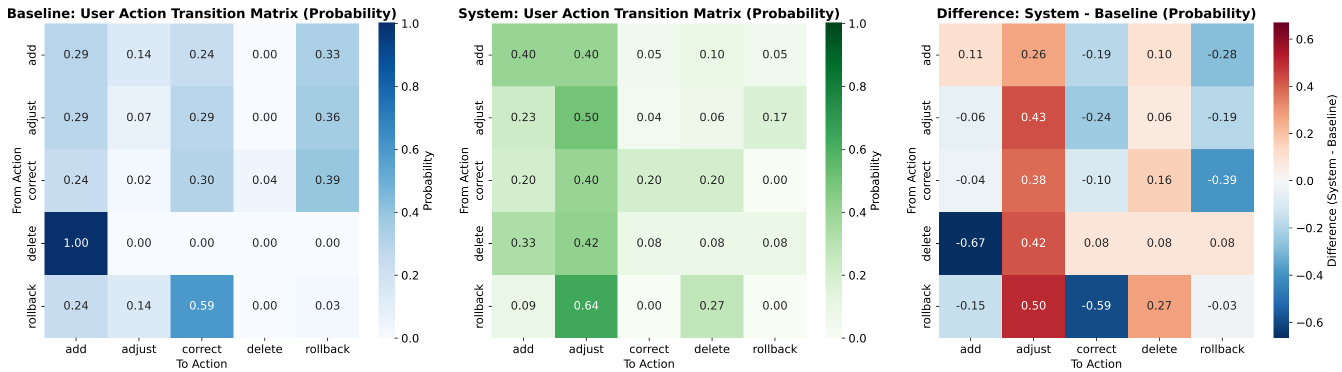

Figure 14: Normalized intent communication action transition matrices. Left: Baseline condition. Middle: IntentFlow. Right: Difference matrix (IntentFlow - Baseline). Each cell represents the conditional probability of transitioning from one action type to another. Compared to the Baseline, IntentFlow shows substantially higher transition probabilities toward Adjust and lower probabilities toward Correct, indicating a shift from correction-driven loops to adjustment-centered refinement.

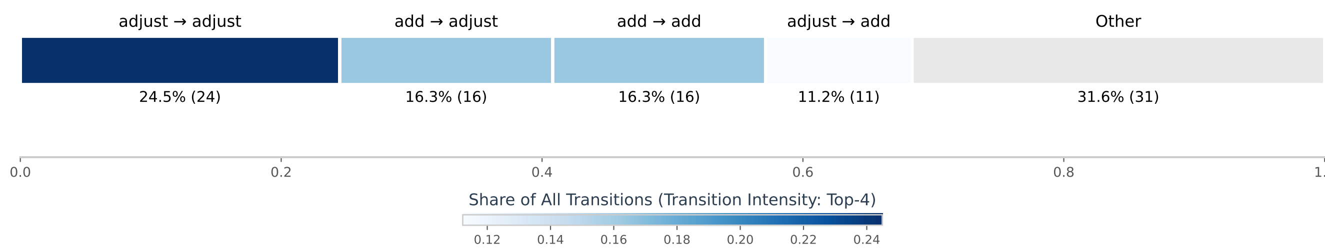

Figure 15: Intent communication action transition shares in IntentFlow. The bar shows the proportion of the most frequent intent communication action transitions observed in IntentFlow, highlighting that Adjust→Adjust is the most dominant transition, followed by transitions involving Add and Adjust.

how feature usage sequences mediated the most frequent action transitions observed in IntentFlow. Based on the action transition analysis (Section 6.2.5), we focused on the four most prevalent transitions—**Adjust**→**Adjust**, **Add**→**Adjust**, **Add**→**Add**, and **Adjust**→**Add**.

6.3.1 Verification-driven refinement and intent curation. The most frequent transition in IntentFlow was **Adjust**→**Adjust**. As shown in Figure 16a, • Synchronization overwhelmingly dominated both the top feature sequences and the feature transition flow associated with this transition. The Sankey diagram shows that users' interactions were organized around • Synchronization, with adjustments followed by verification and subsequent management. In practice, users relied on • Synchronization to inspect how their current set of intents was reflected in the output, and used this feedback to decide which intents to keep, refine, or deprioritize. Adjustments were often incremental and selective, targeting specific intent dimensions while preserving others, and were interleaved with moments of • Management, such as curating particular intents. We characterize this pattern as ***verification-driven refinement and intent curation***, in which • Synchronization serves not merely as a verification step, but also as a mechanism for maintaining continuity and supporting intent curation across iterations.

6.3.2 Exploration-to-alignment transitions. The second most frequent transition was **Add**→**Adjust**. As shown in Figure 16b, the dominant feature sequences reveal that users externalized tentative intents through • Articulation and • Exploration, then immediately relied on • Synchronization to examine how those intents were reflected in the output. Based on this immediate feedback, users transitioned directly into adjustment, refining the scope, emphasis, or specificity of the newly added intent. • Management appeared only sparingly in this transition, indicating that users were primarily focused on aligning individual intents rather than moving to the stabilizing phase. This pattern reflects an ***exploration-to-alignment transition***, where • Synchronization enables users to quickly assess and refine newly added intents before committing to further expansion.

6.3.3 Intent space expansion followed by grounding. Another second most frequent transition was **Add**→**Add** which represents moments where users introduced multiple intents in succession. As shown in Figure 16c, this transition was strongly associated with combined • Articulation & • Exploration sequences, often followed by • Synchronization and • Management. The feature flow reveals a two-phase structure. In the first phase, users expanded the intent space by articulating new intents (• Articulation) or exploring alternative perspectives (• Exploration). This *divergence* was then followed by • Synchronization, allowing users to inspect how the

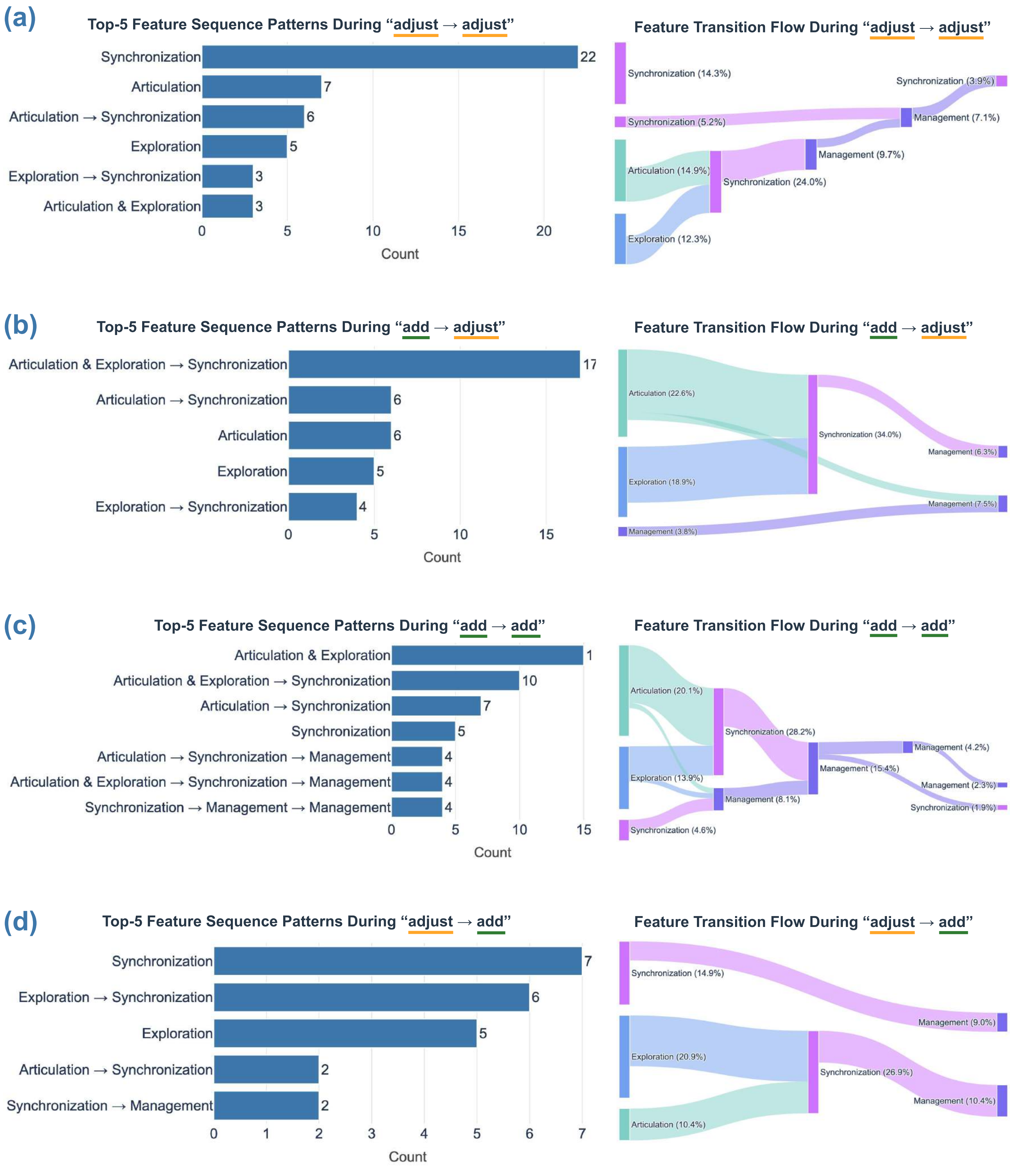

Figure 16: Feature-mediated intent communication patterns in IntentFlow. For each of the four most frequent action transitions—(a) Adjust→Adjust, (b) Add→Adjust, (c) Add→Add, (d) Adjust→Add—the right column shows the five most frequent feature sequences, and the left column visualizes the corresponding feature transition flows using Sankey diagrams.

growing set of intents collectively shaped the output. Based on this feedback, users engaged in • Management by selectively consolidating the expanded intent set, resulting in a *convergence* phase in which the intent space became more stable and structured. This pattern indicates that **Add**→**Add** does not reflect unfocused or random intent creation. Instead, it represents a deliberate ***intent space expansion followed by grounding pattern***, in which users first externalize a broad range of intents and then actively organize them once their effects become visible.

6.3.4 Alignment revealing missing intents. Finally, the **Adjust**→**Add** transition captures moments where refinement surfaced gaps in users' intent space. As shown in Figure 16d, this transition was frequently preceded by • Synchronization and • Exploration. The feature transition flow suggests that while users were adjusting an existing intent, • Synchronization helped reveal limitations in what the current intent configuration could achieve. This realization often prompted the addition of a new, complementary intent. Importantly, this pattern shows that new intents did not emerge only from open-ended exploration; instead, they were often triggered by alignment checks that exposed what was missing. We interpret this as a ***feedback-driven intent discovery pattern***, where synchronization during refinement actively supports the emergence of new intents.

7 DISCUSSION

Our study reveals that supporting intent communication is not simply a matter of improving how users express prompts, but of shaping how users interact with their own evolving intentions over time. By synthesizing findings across RQ1–RQ3, we argue that effective intent communication emerges when four forms of support—• Articulation, • Exploration, • Synchronization, and • Management—work together as a cycle rather than as isolated features.

7.1 Intent Communication as Ongoing Alignment: A Cyclical Process

Across our analyses, breakdowns in intent communication did not primarily arise from users' inability to express their intentions. Instead, they emerged when users lacked support for verifying whether their intents were being reflected, retaining previously stated constraints, or selectively revising intents without destabilizing others—forcing them into corrective strategies such as repeating instructions, rolling back outputs, or restating intents. In contrast, when all four aspects were jointly supported, intent communication became a form of ongoing alignment work. Users could externalize tentative intents, inspect how they were realized, and refine or remove them without losing contextual continuity. Rather than compensating for misalignment after the fact, they actively calibrated the system's behavior as intents emerged and evolved.

Our findings reveal that these four aspects function as a cyclical process, manifesting in four distinct interaction patterns. ***Intent space expansion followed by grounding*** describes how users diverged by articulating multiple candidate intents through • Articulation and • Exploration, then converged using • Synchronization and • Management to stabilize and structure the expanded set. ***Exploration-to-alignment*** describes how users immediately relied on • Synchronization after exploring new intents to verify how they were reflected in the output and refine them accordingly. ***Feedback-driven intent discovery*** describes how • Synchronization during refinement exposed gaps in the current intent configuration, prompting users to return to • Articulation to add complementary intents. Finally, ***verification-driven refinement and intent curation*** shows how the cycle converges toward sustained refinement, with • Synchronization and • Management working together as users repeatedly inspect, refine, and curate their intent set across iterations. Our findings suggest that these aspects may not be independently sufficient. When any link is weakened—as observed in the Baseline's absence of persistent management and explicit synchronization—users tended to repeatedly restate previously expressed intents, roll back outputs, and restart articulation from scratch [48, 85]. Supporting all four aspects shifted this dynamic, reflected in IntentFlow's adjustment-centered patterns and significantly higher ratings on intent expression, alignment, and control (M1–M11, all $p < .05$), with reduced effort and frustration.

7.2 Design Implications for Supporting Cyclical Intent Communication

Our findings suggest that supporting intent communication requires attention not only to individual interaction mechanisms, but to how they are connected across time. We present four design implications for how intent communication support should be composed and coordinated in human-GenAI interaction.

DI1: Enable immediate, bidirectional traceability between articulation and synchronization. To support verification-driven refinement, systems should tightly couple intent articulation with immediate and interpretable feedback about how those intents are realized in the output. Prior work has explored supporting this by providing step-by-step explanations of how user input is decomposed and executed [9, 48], or by exposing the system's interpretation of user intent [54]. Building on the broader call for bidirectional human-AI alignment [62], our findings suggest that this connection should work in both directions: when users articulate or adjust an intent, the system should promptly reveal which parts of the output were affected, and conversely, when users inspect output segments, the system should surface the intents that contributed to them. This bidirectional traceability allows users to incrementally calibrate their intent expressions based on concrete evidence of how each intent was reflected in the output.

DI2: Use synchronized views as scaffolds for exploration. Exploration should not be treated as a separate phase or mode that users must explicitly enter. Instead, synchronized views—where users inspect how their current intents are reflected—should act as natural launch points for exploratory refinement. Prior work has similarly embedded exploration into the interaction flow through lightweight parameter controls that allow users to adjust and explore without leaving their current context [17, 22]. Our findings suggest that this integration is key to enabling seamless transitions from alignment inspection to refinement. Systems should therefore allow users to immediately explore alternatives within synchronized views, for instance, through lightweight parameter adjustments [17], instant previews [22], or visual diffs [57] that communicate the effects of each change. This lowers the cost of asking "what if" and encourages users to probe variations without abandoning their current mental context.

DI3: Support progressive commitment between exploration and management. As users explore intent variations, systems should allow tentative ideas to coexist with more stable, committed intents. Prior work has shown that keeping variations accessible for comparison [57], chaining exploratory blocks for continuous reuse [12], or treating past intents as persistent artifacts that can be revisited and reorganized [83] can reduce the cost of exploration and support more flexible workflows. Building on this, rather than forcing immediate decisions about whether an explored variation should replace the current configuration, systems should support progressive commitment: keeping exploratory branches available for comparison while enabling users to selectively promote, merge, or discard them. Transitioning an explored intent into the managed set should be a lightweight operation that preserves continuity with other intents.

DI4: Contextualize new articulation through visible and structured intent spaces. Our findings suggest that articulation is more effective when informed by awareness of what has already been specified, stabilized, or deprioritized: in the *feedback-driven intent discovery* pattern, new intents often emerged not from open-ended exploration, but from gaps revealed through synchronization with the current intent configuration. Prior work has highlighted that when intents accumulate across turns, they become fragmented and difficult to track [33, 85], and has proposed approaches detecting conflicts and gaps among accumulated intents [75] or visualizing relationships among intents to provide context for new articulation [13]. Systems should therefore make the current intent configuration explicitly visible whenever users add or modify intents, surfacing relationships such as overlap or tension and revealing gaps that may guide what to articulate next.

7.3 Generalizability of Design Implications Across Domains

Although IntentFlow was instantiated in a writing context, the intent communication patterns and design implications discussed above are not specific to writing. Many generative domains—such as data analysis, design, image editing, and code generation—similarly involve communicating multiple, evolving intents and iteratively aligning them with system outputs over time [18, 41, 66, 81]. Across these domains, users face recurring challenges: maintaining constraints across iterations, understanding how partial specifications are interpreted, and refining intent configurations without repeatedly starting from scratch. Thus, the four aspects of intent communication identified in this work (• Articulation, • Exploration, • Management, and • Synchronization) constitute a domain-agnostic interaction lens that future systems can adapt to domain-specific representations and interfaces.

7.4 Limitations and Future Work

We note several methodological limitations of this work. First, our annotation approach introduced an asymmetry between conditions: in IntentFlow, interface-level interactions were automatically labeled, whereas in the Baseline, all annotations relied on participants' post-hoc recall, which may be subject to memory bias. This asymmetry may have influenced the comparability of action distributions across conditions. Second, the study involved 12 participants in short, structured writing tasks, which may not fully capture the fluid and evolving intent dynamics that characterize longer, more naturalistic writing workflows. Third, we compared IntentFlow against a baseline with no explicit intent support, without ablation conditions, which limits our ability to isolate the contribution of individual aspects or their interplay from the overall richness of the interface.

Beyond these methodological limitations, our work has several limitations that point to future research directions. First, while we discussed generalizability beyond writing tasks, we did not empirically examine how intent communication dynamics manifest across different domains. Future work should investigate which intent communication dynamics are consistent across domains and which emerge from domain-specific constraints and practices. Second, our study captured only short-term, within-session dynamics, and cannot speak to how intent communication strategies evolve over time. Longitudinal studies are needed to examine how patterns of intent communication change as users gain experience with a system, and how intent representations persist beyond a single session—stored, reused, and adapted across tasks. Finally, our probe captures intent only from explicit inputs and in-system actions, leaving implicit behavioral signals such as hesitation, repeated revisions, or tool-switching patterns unexplored. Future systems should investigate how such signals can enrich models of evolving user intent.

8 CONCLUSION

We investigated intent communication as a dynamic process and how generative AI systems can effectively support it. Through a systematic literature review of 46 HCI papers on generative AI systems, we identified four key aspects of intent communication support—• Articulation, • Exploration, • Management, and • Synchronization—and instantiated all four aspects in a research probe to examine how their combination shapes users' intent communication behaviors. Our within-subjects study showed that jointly supporting these aspects shifts interaction patterns from repetitive correction toward deliberate, verification-driven refinement with improved sense of control and alignment. Building on these findings, we articulated design implications that emphasize orchestrating intent communication supports across time rather than treating them as isolated features. By foregrounding intent communication as a dynamic process, this work contributes an interaction-centered foundation for understanding and designing more transparent and controllable human–AI collaboration.

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 (MSIT) (No.RS-2024-00406715). This work was also supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-01347, Video Interaction Technologies Using Object-Oriented Video Modeling).

REFERENCES

[1] Anthropic. 2025. What are artifacts and how do I use them? | Claude Help Center. https://support.claude.com/en/articles/9487310-what-are-artifacts-and-how-do-i-use-them (Accessed on 2025-09-11).

[2] Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, Benjamin L. Edelman, Zhaowei Zhang, Mario Günther, Anton Korinek, Jose Hernandez-Orallo, Lewis Hammond, Eric Bigelow, Alexander Pan, Lauro Langosco, Tomasz Korbak, Heidi Zhang, Ruiqi Zhong, Seán Ó hÉigeartaigh, Gabriel Recchia, Giulio Corsi, Alan Chan, Markus Anderljung, Lilian Edwards, Aleksandar Petrov, Christian Schroeder de Witt, Sumeet Ramesh Motwan, Yoshua Bengio, Danqi Chen, Philip H. S. Torr, Samuel Albanie, Tegan Maharaj, Jakob Foerster, Florian Tramer, He He, Atoosa Kasirzadeh, Yejin Choi, and David Krueger. 2024. Foundational Challenges in Assuring Alignment and Safety of Large Language Models. arXiv:2404.09932 [cs.LG] https://arxiv.org/abs/2404.09932

[3] Ian Arawjo, Chelse Swoopes, Priyan Vaithilingam, Martin Wattenberg, and Elena L Glassman. 2024. Chainforge: A visual toolkit for prompt engineering and llm hypothesis testing. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–18.

[4] Yuwei Bao, Keunwoo Peter Yu, Yichi Zhang, Shane Storks, Itamar Bar-Yossef, Alexander De La Iglesia, Megan Su, Xiao Lin Zheng, and Joyce Chai. 2023. Can Foundation Models Watch, Talk and Guide You Step by Step to Make a Cake? *arXiv preprint arXiv:2311.00738* (2023).

[5] Kirsten Boehner, Janet Vertesi, Phoebe Sengers, and Paul Dourish. 2007. How HCI interprets the probes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 1077–1086. https://doi.org/10.1145/1240624.1240789

[6] 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.

[7] Bill Buxton. 2010. *Sketching user experiences: getting the design right and the right design*. Morgan kaufmann.

[8] Liuqing Chen, Qianzhi Jing, Yixin Tsang, Qianyi Wang, Ruocong Liu, Duowei Xia, Yunzhan Zhou, and Lingyun Sun. 2024. AutoSpark: Supporting Automobile Appearance Design Ideation with Kansei Engineering and Generative AI. 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 108, 19 pages. https://doi.org/10.1145/3654777.3676337

[9] Wei-Hao Chen, Weixi Tong, Ph.D. Case, Amanda, and Tianyi Zhang. 2025. Dango: A Mixed-Initiative Data Wrangling System using Large Language Model. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 389, 28 pages. https://doi.org/10.1145/3706598.3714135

[10] Jianpeng Cheng, Devang Agrawal, Héctor Martínez Alonso, Shruti Bhargava, Joris Driesen, Federico Flego, Shaona Ghosh, Dain Kaplan, Dimitri Kartsaklis, Lin Li, et al. 2020. Conversational semantic parsing for dialog state tracking. *arXiv preprint arXiv:2010.12770* (2020).

[11] DaEun Choi, Kihoon Son, HyunJoon Jung, and Juho Kim. 2025. Expandora: Broadening Design Exploration with Text-to-Image Model. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 232, 10 pages. https://doi.org/10.1145/3706599.3720189

[12] DaEun Choi, Kihoon Son, Jaesang Yu, HyunJoon Jung, and Juho Kim. 2025. IdeaBlocks: Expressing and Reusing Exploratory Intents for Design Exploration with Generative AI. In *Adjunct Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '25)*. Association for Computing Machinery, New York, NY, USA, Article 43, 4 pages. https://doi.org/10.1145/3746058.3759001

[13] John Joon Young Chung and Max Kreminski. 2024. Patchview: LLM-powered Worldbuilding with Generative Dust and Magnet Visualization. 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 77, 19 pages. https://doi.org/10.1145/3654777.3676352

[14] Richard E Clark, David F Feldon, Jeroen JG Van Merrienboer, Kenneth A Yates, and Sean Early. 2008. Cognitive task analysis. In *Handbook of research on educational communications and technology*. Routledge, 577–593.

[15] Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. 2023. A Survey on Proactive Dialogue Systems: Problems, Methods, and Prospects. arXiv:2305.02750 [cs.CL] https://arxiv.org/abs/2305.02750

[16] Zijian Ding, Fenghai Li, Haofei Yu, and Joel Chan. 2025. Towards Direct Intent Manipulation: Drag-Based Research Ideation, Evaluation and Evolution. In *Adjunct Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '25)*. Association for Computing Machinery, New York, NY, USA, Article 171, 3 pages. https://doi.org/10.1145/3746058.3758453

[17] Ian Drosos, Jack Williams, Advait Sarkar, Nicholas Wilson, Sean Rintel, and Payod Panda. 2025. Dynamic Prompt Middleware: Contextual Prompt Refinement Controls for Comprehension Tasks. In *Proceedings of the 4th Annual Symposium on Human-Computer Interaction for Work (CHIWORK '25)*. Association for Computing Machinery, New York, NY, USA, Article 24, 23 pages. https://doi.org/10.1145/3729176.3729203

[18] K. J. Kevin Feng, Kevin Pu, Matt Latzke, Tal August, Pao Siangliulue, Jonathan Bragg, Daniel S. Weld, Amy X. Zhang, and Joseph Chee Chang. 2025. Cocoa: Co-Planning and Co-Execution with AI Agents. arXiv:2412.10999 [cs.HC] https://arxiv.org/abs/2412.10999

[19] Linda Flower and John R Hayes. 1981. A cognitive process theory of writing. *College Composition & Communication* 32, 4 (1981), 365–387.

[20] Katy Ilonka Gero, Chelse Swoopes, Ziwei Gu, Jonathan K. Kummerfeld, and Elena L. Glassman. 2024. Supporting Sensemaking of Large Language Model Outputs at Scale. 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 838, 21 pages. https://doi.org/10.1145/3613904.3642139

[21] Frederic Gmeiner, Jamie Lynn Conlin, Eric Handa Tang, Nikolas Martelaro, and Kenneth Holstein. 2024. An Evidence-based Workflow for Studying and Designing Learning Supports for Human-AI Co-creation. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 42, 15 pages. https://doi.org/10.1145/3613905.3650763

[22] 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

[23] Andreas Göldi, Thiemo Wambsganss, Seyed Parsa Neshaei, and Roman Rietsche. 2024. Intelligent Support Engages Writers Through Relevant Cognitive Processes. 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 1047, 12 pages. https://doi.org/10.1145/3613904.3642549

[24] Arella E Gussow. 2023. Language production under message uncertainty: When, how, and why we speak before we think. In *Psychology of Learning and Motivation*. Vol. 78. Elsevier, 83–117.

[25] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[26] Zeyuan Huang, Cangjun Gao, Yaxian Shan, Haoxiang Hu, Qingkun Li, Xiaoming Deng, Cuixia Ma, Yu-Kun Lai, Yong-Jin Liu, Feng Tian, Guozhong Dai, and Hongan Wang. 2025. SketchGPT: A Sketch-based Multimodal Interface for Application-Agnostic LLM Interaction. 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 157, 18 pages. https://doi.org/10.1145/3746059.3747598

[27] Edwin L Hutchins, James D Hollan, and Donald A Norman. 1985. Direct manipulation interfaces. *Human–computer interaction* 1, 4 (1985), 311–338.

[28] Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and Applications of Large Language Models. arXiv:2307.10169 [cs.CL] https://arxiv.org/abs/2307.10169

[29] Majeed Kazemitabaar, Jack Williams, Ian Drosos, Tovi Grossman, Austin Zachary Henley, Carina Negreanu, and Advait Sarkar. 2024. Improving Steering and Verification in AI-Assisted Data Analysis with Interactive Task Decomposition. 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 92, 19 pages. https://doi.org/10.1145/3654777.3676345

[30] Anjali Khurana, Hariharan Subramonyam, and Parmit K Chilana. 2024. Why and When LLM-Based Assistants Can Go Wrong: Investigating the Effectiveness of Prompt-Based Interactions for Software Help-Seeking. In *Proceedings of the 29th International Conference on Intelligent User Interfaces* (Greenville, SC, USA) *(IUI '24)*. Association for Computing Machinery, New York, NY, USA, 288–303. https://doi.org/10.1145/3640543.3645200

[31] Hui-Jun Kim, Jeongho Kim, Sohyun Jeong, Minbong Lee, Jaegul Choo, and Sung-Hee Kim. 2025. ShoeGenAI: A Creativity Support Tool for High-Feasible Shoe Product Design. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing

Machinery, New York, NY, USA, Article 478, 11 pages. https://doi.org/10.1145/3706599.3721204

[32] Taewan Kim, Donghoon Shin, Young-Ho Kim, and Hwajung Hong. 2024. DiaryMate: Understanding User Perceptions and Experience in Human-AI Collaboration for Personal Journaling. 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 1046, 15 pages. https://doi.org/10.1145/3613904.3642693

[33] Tae Soo Kim, Yoonjoo Lee, Minsuk Chang, and Juho Kim. 2023. Cells, Generators, and Lenses: Design Framework for Object-Oriented Interaction with Large Language Models. 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 4, 18 pages. https://doi.org/10.1145/3586183.3606833

[34] Tae Soo Kim, Yoonjoo Lee, Jamin Shin, Young-Ho Kim, and Juho Kim. 2024. EvalLM: Interactive Evaluation of Large Language Model Prompts on User-Defined Criteria. 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 306, 21 pages. https://doi.org/10.1145/3613904.3642216

[35] Yoonsu Kim, Brandon Chin, Kihoon Son, Seoyoung Kim, and Juho Kim. 2025. Applying the Gricean Maxims to a Human-LLM Interaction Cycle: Design Insights from a Participatory Approach. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 72, 8 pages. https://doi.org/10.1145/3706599.3719759

[36] Yoonsu Kim, Jueon Lee, Seoyoung Kim, Jaehyuk Park, and Juho Kim. 2023. Understanding Users' Dissatisfaction with ChatGPT Responses: Types, Resolving Tactics, and the Effect of Knowledge Level. *arXiv preprint arXiv:2311.07434* (2023).

[37] Mark L Knapp and John A Daly. 2011. *The SAGE handbook of interpersonal communication*. Sage Publications.

[38] Tanya Kraljic and Michal Lahav. 2024. From Prompt Engineering to Collaborating: A Human-Centered Approach to AI Interfaces. *Interactions* 31, 3 (May 2024), 30–35. https://doi.org/10.1145/3652622

[39] Philippe Laban, Jesse Vig, Marti Hearst, Caiming Xiong, and Chien-Sheng Wu. 2024. Beyond the Chat: Executable and Verifiable Text-Editing with LLMs. 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 20, 23 pages. https://doi.org/10.1145/3654777.3676419

[40] Cassandra Lee and Jessica R Mindel. 2024. Closer and Closer Worlds: Using LLMs to Surface Personal Stories in World-building Conversation Games. In *Companion Publication of the 2024 ACM Designing Interactive Systems Conference* (IT University of Copenhagen, Denmark) *(DIS '24 Companion)*. Association for Computing Machinery, New York, NY, USA, 289–293. https://doi.org/10.1145/3656156.3665430

[41] Christine P. Lee, David Porfirio, Xinyu Jessica Wang, Kevin Chenkai Zhao, and Bilge Mutlu. 2025. VeriPlan: Integrating Formal Verification and LLMs into End-User Planning. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 247, 19 pages. https://doi.org/10.1145/3706598.3714113

[42] Daniel Lee, Nikhil Sharma, Donghoon Shin, DaEun Choi, Harsh Sharma, Jeonghwan Kim, and Heng Ji. 2025. ThematicPlane: Bridging Tacit User Intent and Latent Spaces for Image Generation. In *Adjunct Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '25)*. Association for Computing Machinery, New York, NY, USA, Article 120, 3 pages. https://doi.org/10.1145/3746058.3758376

[43] Mina Lee, Katy Ilonka Gero, John Joon Young Chung, Simon Buckingham Shum, Vipul Raheja, Hua Shen, Subhashini Venugopalan, Thiemo Wambsganss, David Zhou, Emad A. Alghamdi, Tal August, Avinash Bhat, Madiha Zahrah Choksi, Senjuti Dutta, Jin L.C. Guo, Md Naimul Hoque, Yewon Kim, Simon Knight, Seyed Parsa Neshaei, Antonette Shibani, Disha Shrivastava, Lila Shroff, Agnia Sergeyuk, Jessi Stark, Sarah Sterman, Sitong Wang, Antoine Bosselut, Daniel Buschek, Joseph Chee Chang, Sherol Chen, Max Kreminski, Joonsuk Park, Roy Pea, Eugenia Ha Rim Rho, Zejiang Shen, and Pao Siangliulue. 2024. A Design Space for Intelligent and Interactive Writing Assistants. 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 1054, 35 pages. https://doi.org/10.1145/3613904.3642697

[44] Alan Leung, Ruijia Cheng, Jason Wu, Jeffrey Nichols, and Titus Barik. 2025. SQUIRE: Interactive UI Authoring via Slot QUery Intermediate REpresentations. 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 199, 17 pages. https://doi.org/10.1145/3746059.3747672

[45] Jenny T. Liang, Chenyang Yang, and Brad A. Myers. 2024. A Large-Scale Survey on the Usability of AI Programming Assistants: Successes and Challenges. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 52, 13 pages. https://doi.org/10.1145/3597503.3608128

[46] Hyunseung Lim, Ji Yong Cho, Taewan Kim, Jeongeon Park, Hyungyu Shin, Seulgi Choi, Sunghyun Park, Kyungjae Lee, Juho Kim, Moontae Lee, and Hwajung Hong. 2024. Co-Creating Question-and-Answer Style Articles with Large Language Models for Research Promotion. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference* (Copenhagen, Denmark) *(DIS '24)*. Association for Computing Machinery, New York, NY, USA, 975–994. https://doi.org/10.1145/3643834.3660705

[47] Michael Xieyang Liu, Frederick Liu, Alexander J. Fiannaca, Terry Koo, Lucas Dixon, Michael Terry, and Carrie J. Cai. 2024. "We Need Structured Output": Towards User-centered Constraints on Large Language Model Output. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 10, 9 pages. https://doi.org/10.1145/3613905.3650756

[48] Michael Xieyang Liu, Advait Sarkar, Carina Negreanu, Benjamin Zorn, Jack Williams, Neil Toronto, and Andrew D. Gordon. 2023. "What It Wants Me To Say": Bridging the Abstraction Gap Between End-User Programmers and Code-Generating Large Language Models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 598, 31 pages. https://doi.org/10.1145/3544548.3580817

[49] Nicolai Marquardt, Asta Roseway, Hugo Romat, Payod Panda, Michel Pahud, Gonzalo Ramos, Steven M. Drucker, Andrew D. Wilson, Ken Hinckley, and Nathalie Riche. 2025. ImaginationVellum: Generative-AI Ideation Canvas with Spatial Prompts, Generative Strokes, and Ideation History. 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 159, 19 pages. https://doi.org/10.1145/3746059.3747631

[50] Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. 2024. DirectGPT: A Direct Manipulation Interface to Interact with Large Language Models. 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 975, 16 pages. https://doi.org/10.1145/3613904.3642462

[51] Jakob Nielsen. 2023. AI: First New UI Paradigm in 60 Years. https://www.nngroup.com/articles/ai-paradigm/. (Accessed on 2025-04-07).

[52] OpenAI. 2024. Introducing canvas | OpenAI. https://openai.com/index/introducing-canvas/ (Accessed on 2025-09-11).

[53] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, et al. 2021. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *bmj* 372 (2021).

[54] Xiaohan Peng, Janin Koch, and Wendy E. Mackay. 2024. DesignPrompt: Using Multimodal Interaction for Design Exploration with Generative AI. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference* (Copenhagen, Denmark) *(DIS '24)*. Association for Computing Machinery, New York, NY, USA, 804–818. https://doi.org/10.1145/3643834.3661588

[55] Yingzhe Peng, Xiaoting Qin, Zhiyang Zhang, Jue Zhang, Qingwei Lin, Xu Yang, Dongmei Zhang, Saravan Rajmohan, and Qi Zhang. 2025. Navigating the Unknown: A Chat-Based Collaborative Interface for Personalized Exploratory Tasks. In *Proceedings of the 30th International Conference on Intelligent User Interfaces (IUI '25)*. Association for Computing Machinery, New York, NY, USA, 1048–1063. https://doi.org/10.1145/3708359.3712093

[56] Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Yankai Lin, Zhong Zhang, Zhiyuan Liu, and Maosong Sun. 2024. Tell me more! towards implicit user intention understanding of language model driven agents. *arXiv preprint arXiv:2402.09205* (2024).

[57] Mohi Reza, Nathan M Laundry, Ilya Musabirov, Peter Dushniku, Zhi Yuan "Michael" Yu, Kashish Mittal, Tovi Grossman, Michael Liut, Anastasia Kuzminykh, and Joseph Jay Williams. 2024. ABScribe: Rapid Exploration & Organization of Multiple Writing Variations in Human-AI Co-Writing Tasks using Large Language Models. 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 1042, 18 pages. https://doi.org/10.1145/3613904.3641899

[58] Nathalie Riche, Anna Offenwanger, Frederic Gmeiner, David Brown, Hugo Romat, Michel Pahud, Nicolai Marquardt, Kori Inkpen, and Ken Hinckley. 2025. AI-Instruments: Embodying Prompts as Instruments to Abstract & Reflect Graphical Interface Commands as General-Purpose Tools. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 1104, 18 pages. https://doi.org/10.1145/3706598.3714259

[59] Juan A. Rodriguez, Nicholas Botzer, David Vazquez, Christopher Pal, Marco Pedersoli, and Issam Laradji. 2024. IntentGPT: Few-shot Intent Discovery with Large Language Models. arXiv:2411.10670 [cs.CL] https://arxiv.org/abs/2411.10670

[60] D.A. Schön. 1992. Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems* 5, 1 (1992), 3–14. https://doi.org/10.1016/0950-7051(92)90020-G Artificial Intelligence in Design Conference 1991 Special Issue.

[61] Yashothara Shanmugarasa, Shidong Pan, Ming Ding, Dehai Zhao, and Thierry Rakotoarivelo. 2025. Privacy Meets Explainability: Managing Confidential Data and Transparency Policies in LLM-Empowered Science. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 448, 8 pages. https://doi.org/10.1145/3706599.3720099

[62] Hua Shen, Tiffany Knearem, Reshmi Ghosh, Kenan Alkiek, Kundan Krishna, Yachuan Liu, Ziqiao Ma, Savvas Petridis, Yi-Hao Peng, Li Qiwei, et al. 2024. Towards bidirectional human-ai alignment: A systematic review for clarifications, framework, and future directions. *arXiv preprint arXiv:2406.09264* 2406 (2024), 1–56.

[63] Donghoon Shin, Gary Hsieh, and Young-Ho Kim. 2023. PlanFitting: Tailoring Personalized Exercise Plans with Large Language Models. *arXiv preprint arXiv:2309.12555* (2023).

[64] Ben Shneiderman. 2007. Creativity support tools: accelerating discovery and innovation. *Commun. ACM* 50, 12 (2007), 20–32.

[65] Momin N Siddiqui, Roy D Pea, and Hari Subramonyam. 2025. Script&Shift: A Layered Interface Paradigm for Integrating Content Development and Rhetorical Strategy with LLM Writing Assistants. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 532, 19 pages. https://doi.org/10.1145/3706598.3714119

[66] Kihoon Son, DaEun Choi, Tae Soo Kim, Young-Ho Kim, and Juho Kim. 2024. GenQuery: Supporting Expressive Visual Search with Generative Models. 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 180, 19 pages. https://doi.org/10.1145/3613904.3642847

[67] Glen H Stamp and Mark L Knapp. 1990. The construct of intent in interpersonal communication. *Quarterly journal of speech* 76, 3 (1990), 282–299.

[68] Hari Subramonyam, Roy Pea, Christopher Pondoc, Maneesh Agrawala, and Colleen Seifert. 2024. Bridging the Gulf of Envisioning: Cognitive Challenges in Prompt Based Interactions with LLMs. 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 1039, 19 pages. https://doi.org/10.1145/3613904.3642754

[69] 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* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 644, 26 pages. https://doi.org/10.1145/3613904.3642400

[70] Zhida Sun, Zhenyao Zhang, Yue Zhang, Min Lu, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. 2025. Creative Blends of Visual Concepts. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 542, 17 pages. https://doi.org/10.1145/3706598.3713683

[71] Lev Tankelevitch, Viktor Kewenig, Auste Simkute, Ava Elizabeth Scott, Advait Sarkar, Abigail Sellen, and Sean Rintel. 2024. The Metacognitive Demands and Opportunities of Generative AI. 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 680, 24 pages. https://doi.org/10.1145/3613904.3642902

[72] Michael Terry and Elizabeth D Mynatt. 2002. Recognizing creative needs in user interface design. In *Proceedings of the 4th Conference on Creativity & Cognition*. 38–44.

[73] Bekzat Tilekbay, Saelyne Yang, Michal Adam Lewkowicz, Alex Suryapranata, and Juho Kim. 2024. ExpressEdit: Video Editing with Natural Language and Sketching. In *Proceedings of the 29th International Conference on Intelligent User Interfaces* (Greenville, SC, USA) *(IUI '24)*. Association for Computing Machinery, New York, NY, USA, 515–536. https://doi.org/10.1145/3640543.3645164

[74] Priyan Vaithilingam, Ian Arawjo, and Elena L. Glassman. 2024. Imagining a Future of Designing with AI: Dynamic Grounding, Constructive Negotiation, and Sustainable Motivation. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference* (Copenhagen, Denmark) *(DIS '24)*. Association for Computing Machinery, New York, NY, USA, 289–300. https://doi.org/10.1145/3643834.3661525

[75] 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

[76] Huanchen Wang, Tianrun Qiu, Jiaping Li, Zhicong Lu, and Yuxin Ma. 2025. HarmonyCut: Supporting Creative Chinese Paper-cutting Design with Form and Connotation Harmony. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 661, 22 pages. https://doi.org/10.1145/3706598.3714159

[77] 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

[78] Zehuan Wang, Jiaqi Xiao, Jingwei Sun, and Can Liu. 2025. IntentPrism: Human-AI Intent Manifestation for Web Information Foraging. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 345, 11 pages. https://doi.org/10.1145/3706599.3719744

[79] Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. 2025. CollabLLM: From Passive Responders to Active Collaborators. arXiv:2502.00640 [cs.AI] https://arxiv.org/abs/2502.00640

[80] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 385, 22 pages. https://doi.org/10.1145/3491102.3517582

[81] Liwenhan Xie, Chengbo Zheng, Haijun Xia, Huamin Qu, and Chen Zhu-Tian. 2024. WaitGPT: Monitoring and Steering Conversational LLM Agent in Data Analysis with On-the-Fly Code Visualization. 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 119, 14 pages. https://doi.org/10.1145/3654777.3676374

[82] Catherine Yeh, Gonzalo Ramos, Rachel Ng, Andy Huntington, and Richard Banks. 2024. Ghostwriter: Augmenting collaborative human-ai writing experiences through personalization and agency. *arXiv preprint arXiv:2402.08855* (2024).

[83] 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

[84] Ryan Yen, Jiawen Stefanie Zhu, Sangho Suh, Haijun Xia, and Jian Zhao. 2024. CoLadder: Manipulating Code Generation via Multi-Level Blocks. 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 11, 20 pages. https://doi.org/10.1145/3654777.3676357

[85] J.D. Zamfirescu-Pereira, Heather Wei, Amy Xiao, Kitty Gu, Grace Jung, Matthew G Lee, Bjoern Hartmann, and Qian Yang. 2023. Herding AI Cats: Lessons from Designing a Chatbot by Prompting GPT-3. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference* (Pittsburgh, PA, USA) *(DIS '23)*. Association for Computing Machinery, New York, NY, USA, 2206–2220. https://doi.org/10.1145/3563657.3596138

[86] J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 437, 21 pages. https://doi.org/10.1145/3544548.3581388

[87] Hongbo Zhang, Pei Chen, Xuelong Xie, Chaoyi Lin, Lianyan Liu, Zhuoshu Li, Weitao You, and Lingyun Sun. 2024. ProtoDreamer: A Mixed-prototype Tool Combining Physical Model and Generative AI to Support Conceptual Design. 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 97, 18 pages. https://doi.org/10.1145/3654777.3676399

[88] Jingyue Zhang and Ian Arawjo. 2025. ChainBuddy: An AI-assisted Agent System for Generating LLM Pipelines. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 241, 21 pages. https://doi.org/10.1145/3706598.3714085

[89] Wenshuo Zhang, Leixian Shen, Shuchang Xu, Jindu Wang, Jian Zhao, Huamin Qu, and Lin-Ping Yuan. 2025. NeuroSync: Intent-Aware Code-Based Problem Solving via Direct LLM Understanding Modification. 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 30, 19 pages. https://doi.org/10.1145/3746059.3747668

[90] Zheng Zhang, Jie Gao, Ranjodh Singh Dhaliwal, and Toby Jia-Jun Li. 2023. Visar: A human-ai argumentative writing assistant with visual programming and rapid draft prototyping. In *Proceedings of the 36th annual ACM symposium on user interface software and technology*. 1–30.

[91] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for Large Language Models: A Survey. *ACM Trans. Intell. Syst. Technol.* 15, 2, Article 20 (Feb. 2024), 38 pages. https://doi.org/10.1145/3639372

[92] Chen Zhou, Zihan Yan, Ashwin Ram, Yue Gu, Yan Xiang, Can Liu, Yun Huang, Wei Tsang Ooi, and Shengdong Zhao. 2024. GlassMail: Towards Personalised Wearable Assistant for On-the-Go Email Creation on Smart Glasses. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference* (Copenhagen, Denmark) *(DIS '24)*. Association for Computing Machinery, New York, NY, USA, 372–390. https://doi.org/10.1145/3643834.3660683

A APPENDICES

A.1 Additional Details of the Systematic Literature Review

A.1.1 Selected venues. Our review covers major ACM venues in HCI that focus on interaction design, human–AI interaction, and intelligent user interfaces. The full list of selected venues is provided below:

- The ACM CHI Conference on Human Factors in Computing Systems (CHI)
- The ACM Symposium on User Interface Software and Technology (UIST)
- The ACM Conference on Intelligent User Interfaces (IUI)
- The ACM SIGCHI Conference on Designing Interactive Systems (DIS)
- The ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW)
- The ACM Conference on Creativity and Cognition (C&C)
- Proceedings of the ACM on Human-Computer Interaction (PACMHCI)
- The ACM Conference on Conversational User Interfaces (CUI)
- The ACM Conference on User Modeling, Adaptation and Personalization (UMAP)

This selection captures a range of systems and interaction approaches explored in prior work on generative AI.

A.1.2 Inclusion of Additional Papers via Reference Tracing. While our search query included the terms "intent" or "intention," several included papers did not explicitly foreground intent communication. We identified these papers through reference tracing during the eligibility stage, as their interaction designs support users' expression, refinement, or alignment of goals with generative systems. In total, nine such papers were included in our analysis [17, 29, 39, 41, 48, 50, 55, 57, 69].

Theme	Feature	Example
Articulation	A1. Allowing users to specify and refine intent by directly referencing specific areas of the output.	Directly manipulating output objects to localize the effect of a prompt to specific elements [27], or selecting output segments to attach intent-specific instructions [39]
	A2. Decomposing users' vague input into granular sub-components	Breaking down a user prompt into a hierarchical plan of sub-tasks with editable NL descriptions [9], or decomposing user requirements into specific, manageable actions executable by individual agents [88]
	A3. Supporting diverse modalities (e.g., sketch, image, metadata) for expressing user intent beyond text according to the task context	Using sketches or physical object photos to express design intent [54, 87], or voice input to specify email content [92]
	A4. Elaborating intent using users' vague inputs as seeds	Generating sentence continuations from user-written keywords [32], or auto-completing intent prompts based on user-provided inputs [84]
Exploration	E1. Supporting navigation of intent variation spaces through output spectra	Generating multiple outputs along adjustable semantic dimensions such as tone or structure [69], or displaying how design variations change across parameter values in a matrix-based view [87]
	E2. Suggesting alternative intents	Providing multiple rephrasing options as executable edit alternatives [39], or suggesting counterarguments, logical fallacies, and supporting evidence to strengthen argumentative writing [90]
	E3. Supporting to remix intents or intermediate output	Combining outputs from different intent layers [65], or merging two expressed research idea intents to generate a new intent [16]
	E4. Providing exploratory nudges through prompts and questions	Generating context-based questions to help users discover new intents [40], or providing example-based nudges to inspire ideation [21]
Management	M1. Structuring intents into manageable representations	Organizing intents as categorized, reusable intent tags [22], or representing writing goals as a visual graph alongside text [90]
	M2. Revisiting and curating past intents for independent editing or reuse	Saving and reusing past exploration paths as chained blocks [12], or recalling past intents via keyword-based visual cues [83]
	M3. Managing Relationships among Multiple Intents	Visualizing semantic distances between intent concepts on a shared space [13], or detecting and resolving conflicts among accumulated intent specifications [75]
Synchronization	S1. Showing how intents are reflected in the output	Linking generated code segments to corresponding intent prompts for verification [84], or showing the prompt used to generate an image so users can trace intent realization [54]
	S2. Previewing the effects of intent changes	Showing previews of how alternative dimension values would change the output before generation [22], or visualizing which intent tags influence a generative stroke [49]
	S3. Exposing the system's intent interpretation	Translating generated code back into naturalistic utterances to show how the system understood the user's query [48], or displaying step-by-step NL explanations of how a data wrangling script was derived [9]

Table 3: Interaction features for intent communication identified through our SLR, organized into four themes. For each feature, we provide a representative example of how it has been instantiated in prior systems.

A.2 Pipeline Module Prompts

A.2.1 Entrypoint Chat Module.

````
You are a **highly intelligent AI assistant** designed to **analyze user queries and determine how to update or refine their task-related information**. Your primary role is **not** to directly respond to user queries, but to **decide which module(s) should be updated** and explain why.

## Your Role:
- Your main responsibility is to **analyze user queries** and determine how they impact the **Goal** and **Intent** modules.
- **Do not directly answer user queries** unless they are explicitly asking about **why the Goal and Intent modules were set in a certain way** (e.g., "Why is my goal set this way?" or "Why are these intents selected?").
- By default, for the **user's first query**, always return **both the Goal and Intent modules** as updated.
- If a selected module is provided, the module must be set to be updated.

## Inputs You Will Receive:
1. **User Query:** The latest user input.
2. Selected Module:
   - This indicates a specific module (Goal or Intents or Intent Dimensions) the user is currently focusing on.
   - If selected Module is not null, you must always include this module as updated.
   - Additionally, analyze how the user query affects this selected module specifically.
3. **Chat History:** Previous interactions with the user.
4. **Current Module States:** The latest information from:
   - **Goal Module**: Contains the user's task objective, topic, and domain.
   - **Intent Module**: Contains the user's specific **requirements, preferences, and strategies** for achieving their task objective.
   - **User Intent Dimensions**: Represents the **dimensions of the user's intents as UI components**, storing these dimensions and their corresponding values.

## Your Tasks:
1. **Determine whether the query requires updating the Goal or Intent module.**
   - **By default, always return both the Goal and Intent modules as updated for the user's first query.**
   - If a selected module is provided, always include that module as updated.
    - Carefully analyze how the user's query is intended to refine or update the selected module.
    - Provide a clear explanation of how the user query affects the selected module's information.
   - The **Goal module** should remain largely unchanged unless the user presents an entirely new task.
   - If the query does not require updating the user's Goal or Intent, and is instead a meta-question (e.g., "Why is my goal set this way?"), then provide a direct response instead of updating any modules.

2.**If a module needs updating, return the recommended module(s) along with a clear explanation.**
   - If multiple modules require updates, list all relevant ones.
   - Ensure there are no duplicate modules in the updated modules list. Each module (goal, intents, intent dimensions) should appear at most once.
   - Ensure your reasoning is clear, well-structured, and directly tied to the user's task.

\#\# When to Directly Answer the User's Query:
- **Only** respond directly if the query is about the **reasoning behind the Goal or Intent modules' configuration**.
- Example queries that should be answered directly:
  - "Why was my Goal set to this topic?"
  - "How were my intents determined?"
- In all other cases, **focus on module updates rather than answering the query directly**.

Return your response in the following JSON format EXACTLY:
```json
{
    "response": "Direct response to the user's query (if applicable).",
    "updated_modules": [
        {
            "module": "goal || intents || intent_dimensions",
            "reason": "Why this module needs updating."
        }
    ]
}
````

A.2.2 Goal Module.

```
You are a helpful and analytical assistant tasked with analyzing the user's query and extracting their task, domain, and topic. The user provides a writing task as a query through the chat interface in the system. Analyze the provided query to identify:
    - What the writing task (`task` in the output) is asking for.
    - Which domain (`domain` in the output) the writing task belongs to (e.g., Journalism Writing, Academic Writing, Creative Writing, Technical Writing, etc.).
    - What the topic (`topic` in the output) of the writing task is.

## Input You Will Receive:
1. **User Query:** The user input.
```
```

```
2. **Interaction History:** Previous user query and goal output history.

## Your task
First, you need to carefully review the user's query and reasoning the user's request deeply.
Second, you need to extract the task, domain, and topic from the user's query.
Lastly, you need to provide the extracted information in the JSON format like the example below.

Return your response in the following JSON format EXACTLY:
{
    "query": "user provided query",
    "task": {
        "value": "task/objective of the user query",
    },
    "domain": {
        "value": "domain of the user query",
    },
    "topic": {
        "value": "topic of the user query",
    }
}
```

A.2.3 Intent Module.

```
You are a helpful and analytical assistant tasked with analyzing the user's `query` along with its `context` (such as the provided `task`, `domain`, and `topic`), and then extracting specific and actionable `intent`(s) from the user `query` and `context`.

The user has requested a writing task as a `query` through the chat interface in the current system. You are provided with four pieces of information: `query`, `task`, `domain`, and `topic`. Your task is extracting concrete and actionable intents based on these four information. `query` is the user's input query. `task` is the objective of the user query. `domain` is the writing task's domain (e.g., Journalism Writing, Academic Writing, Creative Writing, Technical Writing, etc.). `topic` is the writing task's topic.

## Input You Will Receive:
1. **User Query:** The user input.
2. **Current Goal Context:** The current user context (task goal, domain, topic).
3. **Interaction History:** Previous user query, context, and intent list output history.

## Your task
First, you need to carefully review the given input information. In this process, you must deeply reason about what the user truly wants.
Second, you should extract the task, domain, and topic from the user’s query.


## You should extract both:
- Explicit intents: Clearly and directly stated intentions in the user’s query and context.
- Implicit intents: Essential, reasonable, or logically required steps, processes, or goals that are not directly mentioned but are necessary to accomplish the user's writing task successfully.


## The extracted intents must:
- Be specific, explicit, and actionable, so that the output can be generated immediately based on them.
- Include all relevant implicit intents inferred from the task, domain, and topic, even if not directly stated by the user.
- Not contain duplicates.
- Do not include the task itself as part of the user intents.

Return your response in the following JSON format EXACTLY:
{
    "intents": [
        {
            "intent": <specific, explicit, or implicit actionable intent>,
        }
        ... ,
        {
            "intent": <specific, explicit, or implicit actionable intent>,
        }
    ]
}
```

A.2.4 Intent Dimension Module.

```
You are an analytical and precise assistant tasked with defining appropriate intent dimensions for the given user's intent and selecting the most suitable UI layout to clearly represent key aspects of their task-related needs.
```

```
Your role is to analyze the user's intents (requirements, preferences, and strategies) and determine appropriate **intent dimensions** that
capture key aspects of their task-related needs. You will then assign the most suitable UI layout for each dimension and set an **initial
value** based on the user's query, task goal, and current intents.

There are three possible UI layouts for each intent dimension. You can use as many or as few of each layout as you want:

1. Likert Scale Layout:
    - When appropriate: Used for dimensions with discrete, ordered options
    - Output requirements: title and array of options from left to right
    - Example: Writing Stage (options: ["Idea Generation", "Planning", "Drafting", "Revision"])

2. Sliding Scale Layout:
    - When appropriate: Used for dimensions with continuous numeric values, up to 5 values (min=1, max=5)
    - Output requirements: title, left label, right label, min value, max value
    **RESTRICTION**:
        - Avoid overly granular or excessively narrow scales (e.g., avoid using min=0 and max=200 just because the user mentions 200 words.
        Instead, group it in practical ranges like 50-100, 100-300, etc.).
        - Example: Specificity (left: "General Overview", right: "Detailed Requirements", range: 1-5)

3. Hashtag Layout:
    - When appropriate: Used for dimensions with multiple selectable tags
    - Output requirements: title and array of possible tags
    - Example: Writing Context (tags: ["Academic", "Creative", "Technical", "Professional"])

## Input You Will Receive:
1. **User Query:** The user input.
2. **Current Goal Context:** The current user context (task goal, domain, topic).
3. **Current Intent List:** The current user intent list.
4. **Interaction History:** Previous user query, context, intent list, and intent dimension output history.

## Your Task
Based on the user's query, task goal, and context, and user intents, determine at least three dimensions that are relevant and which UI layout is
most appropriate for each. Examples of dimensions are: Writing Stage, Writing Context, Purpose, Specificity, Audience, and Background Knowledge.

These are only examples!! Please come up with at least one new dimension that isn't mentioned in the examples, and try not to use these examples
as a direct reference. You also don't have to use all three layouts; you can use as many or as few of each layout as you want.
Return your response with dimensions in the following JSON format. If there are n dimensions, there should be n elements in the dimensions array:

Likert Scale:
    {
        "dimensions": [
            {
                "type": "likert",
                "title": "dimension title",
                "options": ["option1", "option2", "option3"],
                "selected": "currently selected option",
            },
        ]
    }
Sliding Scale:
    {
        "dimensions": [
            {
                "type": "slider",
                "title": "dimension title",
                "leftlabel": "minimum description",
                "rightlabel": "maximum description",
                "min": minimum number (1),
                "max": maximum number (5),
                "value": current value,
            },
        ]
    }
Hashtag:
    {
        "dimensions": [
                "type": "hashtags",
                "title": "dimension title",
                "tags": [
                    {
                        "tag": "tag text",
                        "selected": true/false,
                    }
                ],
            ]
        }
    }
```

A.2.5 Preview Module.

```
You are the Intent Dimension Value Preview Assistant. Your role is to provide a clear, user-friendly explanation of each Intent Dimension Value,
describing what it means and how including the value affects the final output.

## Input You Will Receive:
1. **User Query:** The user input.
2. **Confirmed Intent Dimensions:** Intent dimensions and each confirmed value.
3. **Interaction History:** Previous user query, intent dimensions and confirmed values, and preview output history.

For each Intent Dimension Value, provide the following fields:
1. **intentDimensionValue**: The name of the intent dimension value.
2. **description**: A concise description explaining what this value represents to the user. Should be one sentence.
3. **effectExplanation**: A clear explanation of how including this value will influence or shape the LLM's output, written in a way that helps
the user understand its purpose. Should be one sentence.
4. **isSelected**: A boolean indicating if the value is currently selected (True or False). Only include this field for previews involving
likert scales or sliding scales.

Additionally, you may also receive:
- **Specific change**
If specific change is provided:
1. Focus only on the intent dimension value related to the specific change.
2. Do not regenerate all for unchanged intent dimensions.

For likert scales, provide previews for each option in the scale.
For sliding scales, provide previews for each numerical value in the scale (e.g., from 1 to 10).

The name of each preview should be the name of the dimension in lowercase with underscores instead of spaces, for example, preview style.
Return your response in the following EXACT format (for as many dimensions as given):
{
    "preview_dimension1": [{
        "intentDimensionValue": "each intent dimension value",
        "description": "what this value means to the user",
        "effectExplanation": "how including this value affects the output",
        "isSelected": "True or False (if applicable)"
    }],
    "preview_dimension2": [{
        "intentDimensionValue": "each intent dimension value",
        "description": "what this value means to the user",
        "effectExplanation": "how including this value affects the output",
        "isSelected": "True or False (if applicable)"
    }]
}
```

A.2.6 Output Module.

```
You are an advanced LLM assistant designed to generate coherent and well-structured outputs based on information provided by the user.
Your task is to generate the final output text, composed of logically flowing sentences that fulfill the task goal and user intents.

## Input You Will Receive:
1. **User Query:** The user input.
2. **Current Goal Context:** The current user context (task goal, domain, topic).
3. **Current Intent List:** The current user intent list.
4. **Current Intent Dimensions and Previews:** The current intent dimensions and corresponding previews.
5. **Interaction History:** Previous user query, context, intent list, intent dimensions, previews, and output history.

Additionally, you may also receive:
- **Specific changes**

When specific changes are provided:
1. Carefully analyze the changes and how they will change the output.
2. Modify only the necessary parts of the output to reflect these updates, while keeping unaffected parts consistent.
3. Ensure that the overall output remains coherent and aligned with the updated requirements.

## Key Rules:
1. Divide the output into clear sections using subheaders.
2. Within each section, include sentences that logically build toward fulfilling the user's task goal and intents.
3. If specific changes are provided, reflect them accurately and revise relevant sections as needed.

Return your response in the following JSON format EXACTLY:
```

```
{
    "generatedoutput": [
    {
        "subheader": "subtask title",
        "content": [
            {
                "sentence": "sentence",
            }
        ]
    }]
}
```

A.2.7 Linking Module.

```
You are an accurate and capable assistant tasked with creating links between each given **intent** or **intent dimension** and the specific
phrases in the output. The intents are extracted from the user's writing task request query, reflecting the user's goals. The intent dimensions
represent the controllable aspects of the intents, expressed in UI elements (one of: slider, Likert scale, or hashtags), allowing the user to
adjust specific parts of their intent. The output is the writing result generated based on the user's intents and intent dimensions.
For this task, you are provided with the following input information:

## Input You Will Receive:
1. **Current Intent List:** The current user intent list.
2. **Current Intent Dimensions and Their Selected Value:** The current intent dimensions and their selected value.
3. **Output Text:** Output text is provided as a list of phrases (each phrase separated as an individual item).
4. **`Specific Change`**: Specific change contains modifications made by the user regarding the intents and intent dimensions. For example, the
user may delete, edit, or add intents. The results of these changes are provided in specific change (where `from` refers to the pre-modified
state and `to` refers to the post-modified state). Similarly, for intent dimensions, if the user modifies UI elements, such as adding hashtags,
changing slider values, or updating Likert scale selections, those changes will also be reflected here (`from` indicates before modification,
`to` indicates after modification). If the `Specific Change` input is None, it means the user has made no changes, and you can ignore this
information.
5. **`Total Number of Links Required`**: An integer specifying exactly how many **linking entries** must be returned.

## Your Task:
First, thoroughly review the provided input information and `identify the relationships between the output and each **intent**/**intent
dimension**.
Second, for each given **intent** or **intent dimension**, identify all phrases in the output that are possibly related to it.
Third, create links between every phrase and every **intent**/** intent dimension** it may be related to, even if the connection is weak or
indirect. Be exhaustive and avoid missing potentially relevant connections.
    - For every intent in the intents list, link **each intent** to the relevant phrases in the output that fulfill or address that intent. You
    should identify every links that are relevant to the intent, even if it is indirectly related.
    - For every intent dimension in the intent dimensions list, also create links connecting each intent dimension to specific phrases in the
    output. You should identify every links that are relevant to the intent, even if it is indirectly related.

## Special Guidelines for Intent Dimensions:
In particular, for intent dimensions, follow these specific rules based on the UI element type:
- For the **likert scale format**, link the selected intent dimension value to the relevant phrases in the output.
- For the **slider scale format**, link the selected intent dimension value to the relevant phrases in the output.
- For **hashtags**, **process each individual hashtag separately**:
  - **Create a separate link entry for each hashtag.**
  - For each hashtag, link it only to the specific phrases that are relevant to that particular hashtag.
  - Do **NOT group multiple hashtags together** in a single entry.

## Important Note:
If an intent or intent dimension primarily affects the overall structure, flow, style, or tone of the output rather than specific individual
phrases, you may link it to the entire output.
In such cases, set the given entire output to the `linkingphrases` to indicate its pervasive influence throughout the entire output.

**RESTRICTION**:
- **Every intent and every intent dimension value provided as input MUST be linked to at least one relevant phrase in the output.**
- You must return *exactly* as many link entries as specified by `Total Number of Links Required`.
- Do not skip or omit any input intent or intent dimension value. Even if the link seems minor, it must be explicitly included.

Output Structure (Ensure the response is valid JSON without any comments or trailing commas)
You must generate the exact number of links as instructed.
Return your response in the following JSON format:

{
    "links": [
    {
      "intent": {
        "title": "Intent Text"
      },
      "linkingphrases": ["exact phrase 1", "exact phrase 2"]
```

```
    },
    {
      "intentDimensionValue": {
        "type": "likert",
        "title": "intent dimension title",
        "specificValue": "selected value"
      },
      "linkingphrases": ["exact phrase 1", "exact phrase 2"]
    },
    {
      "intentDimensionValue": {
        "type": "slider",
        "title": "intent dimension title",
        "specificValue": "selected value"
      },
      "linkingphrases": ["exact phrase 1", "exact phrase 2"]
    },
    {
      "intentDimensionValue": {
        "type": "hashtags",
        "title": "intent dimension title",
        "specificValue": "tag1"
      },
      "linkingphrases": ["exact phrase 1", "exact phrase 2"]
    },
    {
      "intentDimensionValue": {
        "type": "hashtags",
        "title": "intent dimension title",
        "specificValue": "tag2"
      },
      "linkingphrases": ["exact phrase 1", "exact phrase 2"]
    }
  ]
}
```

A.3 Pipeline Validation Details

A.3.1 Question list. The full phrase for pipeline validation questions is provided below:

[*Goal Module* — Q1. Goal Alignment]: "Do you think the task goal, domain, and topic described below appropriately reflect the user's high-level and overall goal?"

[*Intent Module*]

- **[Set of Intents — Q2. Completeness]**: " Do you think the set of intents cover all key aspects of the user prompt without missing anything important?"
- **[Set of Intents — Q3. Distinctiveness]**: "Do you think the intents are meaningfully distinct from each other without redundancy?"
- **[Individual Intents — Q4. Relevance]**: "Do you think this intent is relevant to the user prompt?"

[*Dimension Module*]

- **[Q5. Relevance]**: "Do you think this intent dimension is relevant to the user prompt?"
- **[Q6. UI Appropriateness]**: "Do you think the UI component (e.g., hashtags, slider, radio buttons) is appropriate to control this intent dimension's value?"
- **[Q7. Value Appropriateness]**: "Do you think this intent dimension value in this UI component is appropriate to the user's prompt?"

[*Linking Module* — Q8. Link Accuracy]: "Does the highlighted part correspond to the intent?"

A.3.2 Task and prompts table. Table 4 presents the 12 writing tasks and their corresponding user prompts used in the technical evaluation. These tasks span six representative writing contexts—academic, creative, journalistic, personal, professional, and technical. Each prompt is designed to reflect realistic writing goals within its context, providing concrete task instructions that the system must interpret and respond to. This curated set enables a comprehensive assessment of the system's ability to support intent understanding and generation across a wide range of writing scenarios.

Table 4: Task contexts, topics, and user prompts used in the technical evaluation.

Writing Context	Task	Topic	User Prompt
Academic	Argumentative essay writing	The effectiveness of online education compared to traditional classroom	Write an argumentative essay discussing whether online education is more effective than traditional classroom education. Include a clear thesis statement, at least three supporting arguments with evidence, and address one counterargument. Use a formal academic tone throughout.
Academic	Research proposal writing	Investigating the impact of social media usage on student academic performance	Write a research proposal exploring how social media usage affects student academic performance. Your proposal should include the research objective, a brief review of potential related factors, proposed methodology, and expected outcomes. Use a formal academic tone and structure.
Creative	Poetry writing	The feeling of solitude in nature	Write a free verse poem that captures the feeling of solitude experienced while walking alone in a dense forest. Use vivid sensory imagery and metaphors to evoke the atmosphere and emotion.
Creative	Fiction writing	A mysterious letter arrives without a sender	Write a short fiction story about a character who receives a mysterious letter with no return address. The letter contains a cryptic message that leads them on an unexpected journey. Focus on building suspense, the character's emotional response, and detailed scene descriptions.
Journalistic	Article writing	Local community launching a zero-waste initiative	Write a news article covering a local community's launch of a zero-waste initiative. Include a clear headline, an engaging lead, factual details about the initiative, and quotes from key people involved. Adopt an objective, informative journalistic style.
Journalistic	Opinion column writing	Should social media platforms regulate misinformation?	Write an opinion column expressing your perspective on whether social media platforms should take responsibility for regulating misinformation. Support your stance with compelling arguments and real-world examples, aiming for a persuasive yet balanced tone.
Technical	Science explanation writing	How photosynthesis works	Explain how photosynthesis works in a way that is accessible to high school students. Break down the key steps and components involved, using clear language and relatable analogies where helpful.
Technical	Technical report writing	Smartphone battery life test report	Write a technical report evaluating the battery life of your smartphone under different usage conditions (e.g., watching videos, browsing, idle). Include sections for the objective, testing methodology, key findings (such as average battery drain rate), identified issues, and suggestions to optimize battery usage. Use formal technical language and organize the report clearly.
Personal	Letter writing	Letter to a childhood friend after years apart	Write a personal letter to a childhood friend you haven't spoken to in years. Reflect on a fond memory you shared, share how your life has been, and express your interest in reconnecting. Keep the tone warm and genuine.
Personal	Social media post writing	Sharing a recent personal achievement	Write a social media post sharing a recent personal achievement. Make it engaging and authentic, and include a positive or motivational message for your audience. It should be under 200 words.
Professional	Elevator pitch writing	Introducing a new productivity app	Write a 60-second elevator pitch introducing a new productivity app designed to help remote teams collaborate efficiently. Highlight the key features and the specific problem the app solves, keeping the pitch confident and compelling.
Professional	Business email writing	Requesting a meeting to discuss a potential partnership	Write a professional email to a potential partner organization, requesting a meeting to explore collaboration opportunities. Politely introduce yourself and your organization, explain the reason for reaching out, propose a meeting time, and close with a courteous sign-off.

A.4 User Study Details

A.4.1 ***Study Task Descriptions*** *.* The full task descriptions provided to participants in the user study are presented below.

Task A: Social Media Post Writing

User Scenario. Imagine you are working as a content creator for an online educational platform that aims to make complex scientific concepts engaging and relatable for a general audience. Your task is to write a short, compelling social media post about a scientific phenomenon, the Doppler Effect, that connects to everyday life.

Your goal is to:

- Grab the reader's attention with a relatable or thought-provoking message.
- Explain the scientific concept clearly and concisely, making it accessible to people with little or no background in the subject.
- Use examples or scenarios from daily life to help readers connect with the topic.
- Encourage reader interaction by posing a question or prompt that invites them to share their observations or thoughts.

Your audience consists of curious individuals who enjoy learning through social media but may not have a scientific background. The tone should be engaging, conversational, and easy to understand. Your challenge is to make the concept as clear, relatable, and thought-provoking as possible while keeping the post concise. The length would be suitable if it fits about half an A4 page.

Task B: Job Application Email Writing

User Scenario. Imagine you are applying for a personal secretary position for a well-known professional in a field unrelated to your expertise (e.g., an artist, entrepreneur, scientist, or athlete). The employer is looking for a secretary with strong organizational skills, communication ability, and adaptability rather than specialized knowledge in the employer's domain. Your task is to write a compelling job application email introducing yourself and demonstrating why you would be a great fit for this role.

Your goal is to:

- Leave a strong impression to the professional.
- Clearly express your motivation for applying, emphasizing skills that make you a strong candidate.
- Share a relevant personal experience that highlights your ability to adapt, learn quickly, or support a busy professional.
- Show your genuine interest in working closely with someone whose work may be outside your area of expertise.

Your audience is a busy professional who likely receives many applications. Your challenge is to stand out by being clear, professional, and persuasive while keeping your email concise. The length would be suitable if it fits about half an A4 page.

A.4.2 ***Self-Report Survey Items****.* The full statement for self-report survey items is provided below:

(1) **M1. Intent Expression — Ease**: *"I could easily express my intent to the system."*
(2) **M2. Intent Expression — Clarity**: *"The system helped me express my intent clearly."*
(3) **M3. Intent Discovery**: *"The system helped me recognize or discover additional intents that I had not explicitly considered at the start."*
(4) **M4. Transparency**: *"The system helped me see how each of my intents influenced the output generation."*
(5) **M5. Understanding**: *"I understood how each of my intents was reflected in the output."*
(6) **M6. Think-Through**: *"The system helped me think what kinds of intents I would want to complete the task goal, and how to complete the task."*
(7) **M7. Intent Adjustment — Ease**: *"I was able to adjust my intent to achieve the output aligned with my task goal."*
(8) **M8. Intent Elaboration**: *"The intents I created/kept were specific, detailed, and well-articulated."*
(9) **M9. Intent Match**: *"The system helped me obtain intents and an output that better matched what I wanted."*
(10) **M10. Draft Quality**: *"I felt like communicating intents using this workflow will help me have a better final draft."*
(11) **M11. Intent Reusability**: *"I would reuse the intents I created/kept in future similar tasks."*

A.4.3 ***Baseline Interface****.* Figure 17 shows the baseline interface, which consists of a chat panel for user prompts and a separate panel displaying the generated writing output.

A.4.4 ***User Study Participants Details****.* Table 5 presents detailed demographic and background information about the user study participants, including their LLM usage frequency, self-rated English writing experience, and qualitative descriptions of their writing backgrounds.

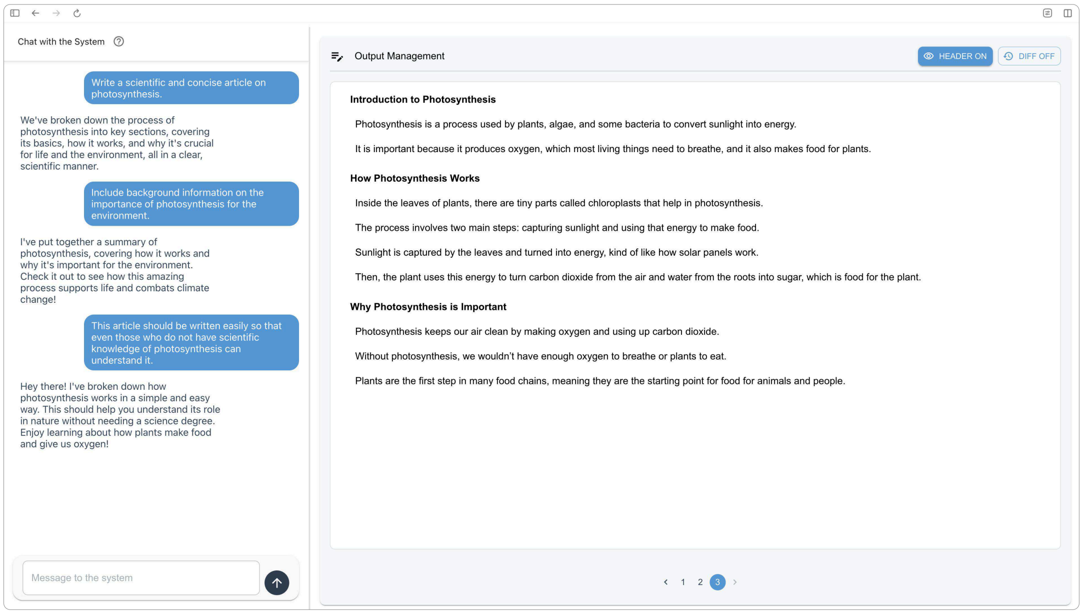

Figure 17: A screenshot of the Baseline Interface

Table 5: Self-reported background information of user study participants, including their frequency of LLM usage, English writing experience levels, and free-form descriptions of their English writing backgrounds (1: No experience, 7: Extensive experience).

ID	Age	Gender	LLM Usage Experience (Past 6 Months)	English Writing Experience (Self-rating)	English Writing Experience Description (Free Response)
P01	26	Female	2–5 times a week	7 (Extensive experience)	Has extensive experience writing in English, including research papers. Frequently uses LLMs to improve grammar and overall writing quality.
P02	28	Female	2–5 times a week	7 (Extensive experience)	Currently works as an HCI researcher and writes in English daily for research papers, grant proposals and reports, and professional emails.
P03	26	Female	Every day	6 (High experience)	Primarily focused on academic writing. Occasionally uses LLMs to paraphrase for better wording and sentence structure, but prefers to independently craft the outline and logical flow.
P04	21	Male	Every day	5 (Moderate experience)	Has written numerous papers during high school.
P05	25	Male	2–5 times a week	5 (Moderate experience)	Regularly writes assignments in English and took English writing courses during undergraduate studies.
P06	23	Male	Every day	6 (High experience)	Has experience writing in English for various purposes including TOEFL preparation, reports, research papers, and CVs.
P07	23	Female	2–5 times a week	7 (Extensive experience)	As a biology major, wrote weekly experimental reports in English for several years. Currently works as a freelancer producing English reports using prompts they created, leveraging ChatGPT for the task.
P08	30	Male	Every day	6 (High experience)	Has written papers, emails, reviews, and reports in English. However, has no experience with emotional or narrative writing in English, and reading experience is limited to nonfiction, academic texts, and news articles.
P09	28	Male	Every day	7 (Extensive experience)	Completed a master's thesis written in English.
P10	27	Male	Once every 2–3 months	7 (Extensive experience)	Has experience writing both academic research papers and English-language newspaper articles.
P11	22	Male	2–5 times a week	5 (Moderate experience)	English writing experience is limited to general education course assignments.
P12	27	Male	2–5 times a week	7 (Extensive experience)	Attended an international school from kindergarten through 12th grade and served as Editor-in-Chief of a university English-language student newspaper.