Feedforward in Generative AI: Opportunities for a Design Space

Bryan Min
bdmin@ucsd.edu
University of California San Diego
La Jolla, California, USA

Haijun Xia
haijunxia@ucsd.edu
University of California San Diego
La Jolla, California, USA

ABSTRACT

Generative AI (GenAI) models have become more capable than ever at augmenting productivity and cognition across diverse contexts. However, a fundamental challenge remains as users struggle to anticipate what AI will generate. As a result, they must engage in excessive turn-taking with the AI's feedback to clarify their intent, leading to significant cognitive load and time investment. Our goal is to advance the perspective that in order for users to seamlessly leverage the full potential of GenAI systems across various contexts, we must design GenAI systems that not only provide informative feedback but also informative feedforward—designs that tell users what AI will generate before the user submits their prompt. To spark discussion on feedforward in GenAI, we designed diverse instantiations of feedforward across four GenAI applications: conversational UIs, document editors, malleable interfaces, and agent automations, and discussed how these designs can contribute to a more rigorous investigation of a design space and a set of guidelines for feedforward in all GenAI systems.

CCS CONCEPTS

• **Human-centered computing** → **Interaction paradigms**; HCI theory, concepts and models.

KEYWORDS

Feedforward, Human-AI Interaction, Generative AI

1 INTRODUCTION

Generative AI (GenAI) models are capable of generating a diverse range of content including stories, videos, code, and more. HCI communities have integrated these models into a diverse range of applications to augment learning and teaching [15, 32], support academic literature review [1, 12, 16, 19], generate and customize software on the fly [6, 17, 24, 36], and automate web tasks [11, 29], showcasing their potential to enhance productivity and cognition on a universal scale.

Despite its potential, users inevitably face challenges in trying to anticipate GenAI's responses and aligning their prompts to match their needs. Subramonyam et al. have labeled this set of challenges as the *gulf of envisioning*, which describes the user's gap in identifying and articulating their desired task, providing effective instructions, and anticipating the model's response [30]. This gap makes it difficult for novices to formulate effective prompts to AI and understand the cause of misalignment [38].

While researchers have explored various approaches to help users better learn and anticipate AI's responses with dedicated spaces for experimenting with prompts [3, 4], generating multiple outputs in a single space [13, 31], asking clarifying questions [28], and providing widgets to modify generated outputs posthoc [14, 23, 36], these systems primarily focus on offering *feedback*—helping users understand what happened *after* they submit a prompt. While feedback helps users develop an understanding of the system through exploration, users cannot always anticipate the outcome of their prompts. Instead, they must rely on iterative trial-and-error, which can incur significant cognitive load and time investment. If we want users to easily leverage the potential of GenAI systems across diverse contexts, they must be able to anticipate AI outputs without solely relying on feedback.

To achieve this, we aim to advance the perspective that GenAI must not only provide informative feedback but also informative *feedforward*—designs that help users anticipate outcomes *before* performing an action [37]. Providing such feedforward in GenAI systems can help users understand what to expect from the AI model before submitting their prompt. This can reduce excessive conversational exchanges, bridge the abstraction gap between users and AI [21, 30, 35], and encourage meta-cognition when constructing prompts [33].

The HCI community has extensively explored feedforward design in various domains, such as gestures [5], VR/AR [25], widgets [8, 34], and proactive systems [2]. Furthermore, the community has begun exploring feedforward designs for GenAI, examining trade-offs between the level of detail provided and the cognitive load involved in assessing generated code and images [10, 39]. However, GenAI's expanding capabilities are shaping an increasingly broad design space of GenAI interfaces. This rapid growth calls for a more systematic approach to designing effective feedforward across diverse domains—uncovering opportunities for novel interaction techniques, assessing user benefits and challenges across design dimensions, and establishing comprehensive design guidelines.

This paper aims to spark discussion on how we may achieve these goals, enabling users to effectively engage and interact with all forms of GenAI systems. To lay the groundwork for this discussion, we designed diverse instantiations of feedforward across four GenAI applications: conversational UIs, document editors, malleable interfaces, and agent automations. We then discussed the
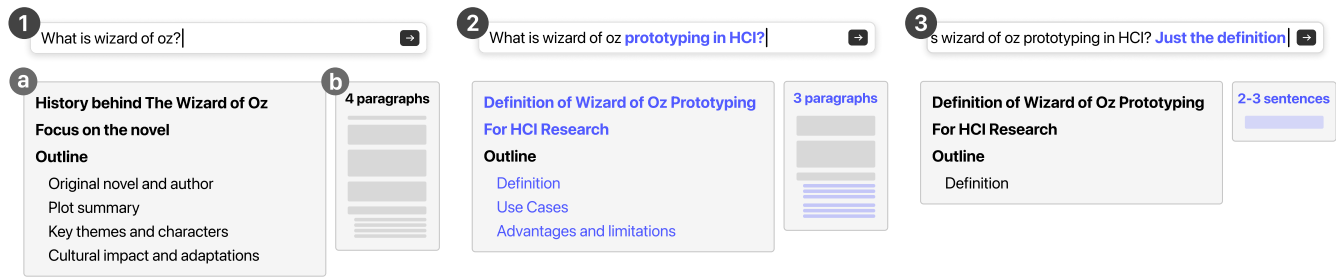
Figure 1: (1) When the user stops typing for a brief moment, the conversational UI presents two feedforward components: (a) a list of key topics and an outline, and (b) a visual minimap of the anticipated length of the response. (2) The user anticipates what the LLM will generate and adjusts their prompt to match it. (3) They also ask for less information, in which the feedforward components update to match the request.
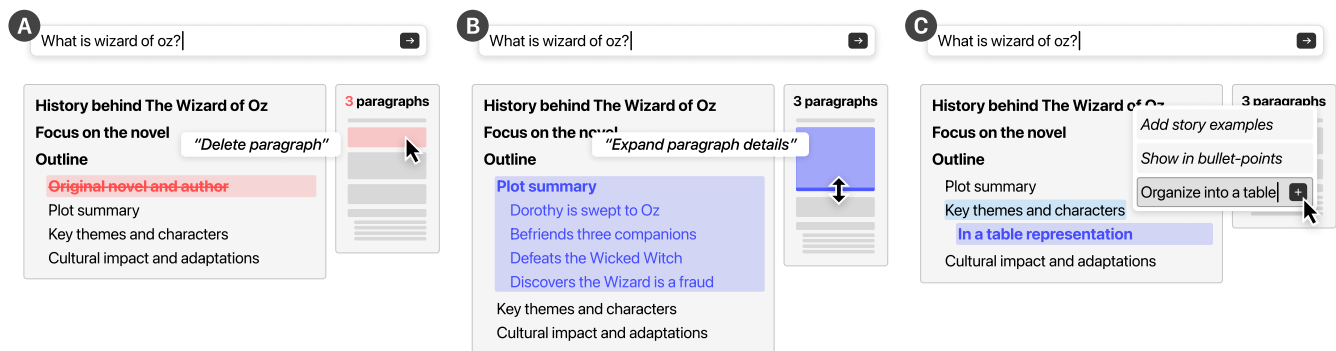

Figure 2: Users can directly interact and manipulate feedforward components by for instance (A) deleting unneeded paragraphs, (B) expanding details about a section, and (C) transforming the representation of the expected content.

designs and implementations needed to further develop a comprehensive design space for informative feedforward GenAI systems.

We invite researchers in this community to critically examine past and present GenAI interfaces through the lens of feedforward, draw from feedforward literature in other domains, explore new interaction techniques and frameworks for delivering informative feedforward, and work toward concrete design and implementation strategies applicable to all GenAI systems.

2 FEEDFORWARD IN GENERATIVE AI

This section aims to explore possible ways to provide feedforward in generative AI systems. We view that effective feedforward in generative AI interfaces should enable users to:

- Anticipate what the AI's response or action may be.
- Disambiguate their prompt without excessive exchanges in conversation or interaction.
- Directly engage with the feedforward content.
- Reflect on their prompting practices with less friction.

To explore various ways to design feedforward in GenAI systems, we first explore feedforward in an LLM-powered conversational interface. We then describe scenarios for three other designs of feedforward in GenAI applications.

2.1 Feedforward in Conversational LLM UIs

2.1.1 Feedforward Components. Research has shown that users benefit from multiple representations when interacting with LLMs [14, 32]. Therefore, we developed feedforward components that each aim to provide a specific representation that would help users anticipate the LLM's response. As the user types a prompt, the conversational UI generates feedforward content inside two feedforward components: 1) a key topic phrase alongside a high-level outline of the anticipated response (Fig. 1.1a) and 2) a minimap view with paragraph blocks (Fig. 1.1b). These representations help users anticipate the topics and subtopics the LLM will generate, as well as whether its response might be too short or too long.

2.1.2 Manipulating Feedforward Content. We enable users to manipulate feedforward content in two ways. First, when users edit their prompt, the feedforward content updates to reflect the changes. This allows users to identify misalignments between their intent and the feedforward content, encouraging them to add details to the prompt before submitting it. For instance, they can clarify an ambiguous term (Fig. 1.2) or request a shorter response (Fig. 1.3). These instant updates to the feedforward components may also spark reflection, such as realizing that it may be useful to understand the advantages and limitations of Wizard of Oz Prototyping rather than just its definition.
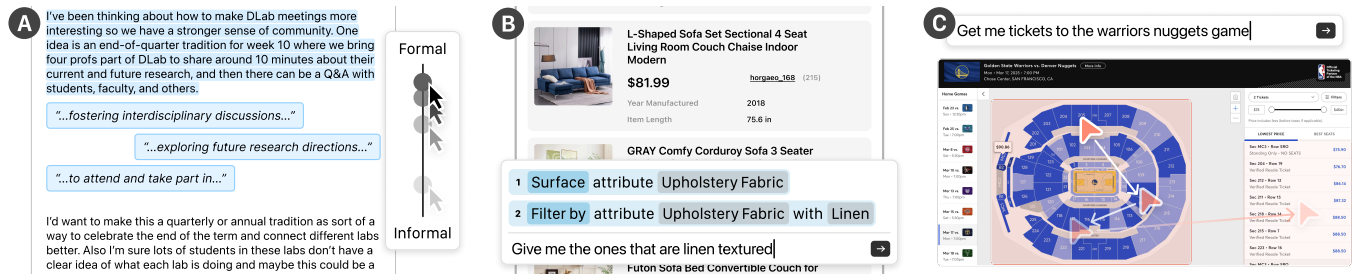
Figure 3: We explored three different applications for designing feedforward in GenAI systems: (A) document editors, (B) malleable interfaces, and (C) agent automation systems.

Second, users can directly edit content inside the feedforward components. For example, rather than prompting the LLM to remove subtopics from the outline, users can directly delete the subtopic title in the outline component or delete the paragraph in the minimap (Fig. 2A). Users can also drag the edge of the minimap component to expand the feedforward content's level of details (Fig. 2B). This can allow users control "how far ahead" they can look into the LLM's response. Additionally, users can highlight parts of the outline, revealing a tooltip that generates a list of possible actions and outputs based on that selection, such as adding examples about a subtopic or organizing the content in a bullet-point list. Users can either select one of these actions or request their own, which will update the feedforward outline (Fig. 2C).

2.2 Application Scenarios

Our goal is to explore opportunities for designing feedforward in all GenAI interfaces—beyond conversational UIs. In this paper, we explore three applications to showcase various feedforward designs across diverse contexts: document editors, malleable interfaces, and agent automations.

2.2.1 Document Editors. Developers and researchers have begun to integrate AI into document editors and writing canvases to support various aspects of writing tasks, such as fixing grammar [18], presenting continuously synchronized outlines [9], and providing widgets to adjust the tone of writing along various dimensions [22, 27]. However, it can be difficult for users to anticipate what the LLM might produce based on their writing and revisions. For instance, moving a slider up to change the tone from "informal" to "formal" does not indicate exactly how "formal" they are making their writing. It is only after submitting the change and reviewing the final result that users realize whether they have achieved the desired outcome. To reduce the amount trial-and-error to achieve the desired result, document editors can provide feedforward that presents example phrases as the user moves the slider up and down, helping the user quickly develop a clear understanding of the level of formality they are adjusting to (Fig. 3A). This feedforward mechanism can be expanded to let users control how much detail is shown in each example phrase, allowing them to reveal more or fewer words as needed. This customization helps align the feedforward component with user preferences, reducing cognitive load.

2.2.2 Malleable Interfaces. GenAI is additionally enabling software interfaces to become increasingly malleable, enabling users to generate custom functional applications on the fly [20, 26], personalized interfaces that blends user activity [6], and user-defined abstractions in overview-detail interfaces [24]. In current generated malleable interfaces, the user prompts for a custom application and provides specifications, and after rounds of clarifying, AI generates and compiles the requested application. However, it can be unclear what kinds of software components the AI might produce based on their prompts alone. For example, if the user asks AI to show only linen-textured couches from a shopping website, the user cannot be sure whether AI will add a filtering operation, perform another search query, or create a brand new list linen-textured couches. Feedforward can present a list of system operations the AI will perform before submitting the prompt, allowing them to quickly assess and revise the operations without unnecessary exchanges with the AI (Fig. 3B).

2.2.3 Agent Automations. The AI community has also presented demonstrations of AI agents that can automate diverse tasks on the web [7, 29]. However, there lacks clear communication of exactly what the agent is doing on the screen. For instance, if a user asks an AI agent to look for tickets to a basketball game, the agent will break the task into steps and perform them by opening and navigating a webpage. During this phase, the user may be unaware of which buttons or components the agent plans to click to find the best prices, which portions of the webpage it will scan for context, or where it will navigate next. Agent automation systems can integrate feedforward visualizations by providing area selections to show where the AI agent is focusing on for context and opaque cursors to tell the user where the agent will click next (Fig. 3C). These feedforward representations help users anticipate the agent's future actions and provide visual affordances to intervene in potential errors by adjusting the selection area or dragging the cursor.

3 FUTURE WORK

Our goal is to establish feedforward as a fundamental design component in all GenAI systems. While this paper presents several feedforward designs, we envision a more comprehensive design space to represent and guide feedforward design across all GenAI applications. Based on our four prototypes, we identify three potential design dimensions for GenAI feedforward: representation, level of detail, and manipulability.

First, our examples implemented feedforward representations in the form of outlines, minimaps, lists of operations, example phrases, and multiple cursors. We may identify categories of representations that help distinguish which feedforward representations are more useful for different use cases. For instance, while a list of operations might inform users about the type of UI a GenAI system will generate, a wireframe might better communicate the structure and layout of the UI.

Second, we explored different ways to present varying levels of detail in feedforward. For example, the conversational UI displays an outline summarizing key topics, while the minimap omits textual details and instead presents blocks of paragraphs. This dimension aligns with previous research on feedforward [5]. A more rigorous investigation could explore optimal levels of detail for different types of feedforward representations in GenAI, as well as allowing users to define the level and type of detail themselves.

Lastly, we explored how users can manipulate feedforward content, either by revising their prompts or by directly resizing, repositioning, or selecting elements. We aim to investigate additional interaction techniques that enhance user control and engagement with feedforward designs.

Future work should expand on this preliminary design space by gathering, analyzing, and critiquing a broader range of GenAI systems to identify variations of feedforward across diverse contexts.

REFERENCES

[1] 2024. Elicit. https://elicit.com/ Accessed April 5, 2024.
[2] J.E. Allen, C.I. Guinn, and E. Horvtz. 1999. Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications* 14, 5 (1999), 14–23. https://doi.org/10.1109/5254.796083
[3] Tyler Angert, Miroslav Suzara, Jenny Han, Christopher Pondoc, and Hariharan Subramonyam. 2023. Spellburst: A Node-based Interface for Exploratory Creative Coding with Natural Language Prompts. 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 100, 22 pages. https://doi.org/10.1145/3586183.3606719
[4] 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* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 304, 18 pages. https://doi.org/10.1145/3613904.3642016
[5] Olivier Bau and Wendy E. Mackay. 2008. OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology* (Monterey, CA, USA) *(UIST '08)*. Association for Computing Machinery, New York, NY, USA, 37–46. https://doi.org/10.1145/1449715.1449724
[6] Yining Cao, Peiling Jiang, and Haijun Xia. 2025. Generative and Malleable User Interfaces with Evolving Task-Driven Data Schema. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '25)*. Association for Computing Machinery, New York, NY, USA, 21 pages. https://doi.org/10.1145/3706598.3713285
[7] The Browser Company. 2025. Come build Dia. https://www.diabrowser.com/ Accessed February 11, 2025.
[8] Sven Coppers, Kris Luyten, Davy Vanacken, David Navarre, Philippe Palanque, and Christine Gris. 2019. Fortunettes: Feedforward about the Future State of GUI Widgets. *Proc. ACM Hum.-Comput. Interact.* 3, EICS, Article 20 (June 2019), 20 pages. https://doi.org/10.1145/3331162
[9] Hai Dang, Karim Benharrak, Florian Lehmann, and Daniel Buschek. 2022. Beyond Text Generation: Supporting Writers with Continuous Automatic Text Summaries. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 98, 13 pages. https://doi.org/10.1145/3526113.3545672
[10] Hai Dang, Lukas Mecke, and Daniel Buschek. 2022. GANSlider: How Users Control Generative Models for Images using Multiple Sliders with and without Feedforward Information. 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 569, 15 pages. https:

//doi.org/10.1145/3491102.3502141
[11] 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
[12] Raymond Fok, Joseph Chee Chang, Tal August, Amy X. Zhang, and Daniel S. Weld. 2024. Qlarify: Recursively Expandable Abstracts for Dynamic Information Retrieval over Scientific Papers. 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 145, 21 pages. https://doi.org/10.1145/3654777.3676397
[13] 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
[14] Peiling Jiang, Jude Rayan, Steven P. Dow, and Haijun Xia. 2023. Graphologue: Exploring Large Language Model Responses with Interactive Diagrams. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 3, 20 pages. https://doi.org/10.1145/3586183.3606737
[15] Hyoungwook Jin, Seonghee Lee, Hyungyu Shin, and Juho Kim. 2024. Teach AI How to Code: Using Large Language Models as Teachable Agents for Programming Education. 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 652, 28 pages. https://doi.org/10.1145/3613904.3642349
[16] Hyeonsu B Kang, Tongshuang Wu, Joseph Chee Chang, and Aniket Kittur. 2023. Synergi: A Mixed-Initiative System for Scholarly Synthesis and Sensemaking. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 43, 19 pages. https://doi.org/10.1145/3586183.3606759
[17] Tae Soo Kim, DaEun Choi, Yoonseo Choi, and Juho Kim. 2022. Stylette: Styling the Web with Natural Language. 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 5, 17 pages. https://doi.org/10.1145/3491102.3501931
[18] 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
[19] Yoonjoo Lee, Hyeonsu B Kang, Matt Latzke, Juho Kim, Jonathan Bragg, Joseph Chee Chang, and Pao Siangliulue. 2024. PaperWeaver: Enriching Topical Paper Alerts by Contextualizing Recommended Papers with User-collected Papers. 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 19, 19 pages. https://doi.org/10.1145/3613904.3642196
[20] Geoffrey Litt. 2025. https://x.com/geoffreylitt/status/1875213046366511516 Accessed February 11, 2025.
[21] 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
[22] Damien Masson, Young-Ho Kim, and Fanny Chevalier. 2024. Textoshop: Interactions Inspired by Drawing Software to Facilitate Text Editing. arXiv:2409.17088 [cs.HC] https://arxiv.org/abs/2409.17088
[23] 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
[24] Bryan Min, Allen Chen, Yining Cao, and Haijun Xia. 2025. Malleable Overview-Detail Interfaces. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '25)*. Association for Computing Machinery, New York, NY, USA, 26 pages. https://doi.org/10.1145/3706598.3714164
[25] Andreea Muresan, Jess Mcintosh, and Kasper Hornbæk. 2023. Using Feedforward to Reveal Interaction Possibilities in Virtual Reality. *ACM Trans. Comput.-Hum. Interact.* 30, 6, Article 82 (Sept. 2023), 47 pages. https://doi.org/10.1145/3603623
[26] onetwoval. 2025. https://x.com/onetwoval/status/1887547069956845634 Accessed February 11, 2025.

[27] OpenAI. 2025. Introducing Canvas. https://openai.com/index/introducing-canvas/ Accessed February 11, 2025.

[28] OpenAI. 2025. Introducing Deep Research. https://openai.com/index/introducing-deep-research/ Accessed February 3, 2025.

[29] OpenAI. 2025. Introducing Operator. https://openai.com/index/introducing-operator/ Accessed February 3, 2025.

[30] 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

[31] 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

[32] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. 2023. Sensecape: Enabling Multilevel Exploration and Sensemaking with Large Language Models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 1, 18 pages. https://doi.org/10.1145/3586183.3606756

[33] 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

[34] Michael Terry and Elizabeth D. Mynatt. 2002. Side views: persistent, on-demand previews for open-ended tasks. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (Paris, France) *(UIST '02)*. Association for Computing Machinery, New York, NY, USA, 71–80. https://doi.org/10.1145/571985.571996

[35] 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

[36] Priyan Vaithilingam, Elena L. Glassman, Jeevana Priya Inala, and Chenglong Wang. 2024. DynaVis: Dynamically Synthesized UI Widgets for Visualization Editing. 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 985, 17 pages. https://doi.org/10.1145/3613904.3642639

[37] Jo Vermeulen, Kris Luyten, Elise van den Hoven, and Karin Coninx. 2013. Crossing the bridge over Norman's Gulf of Execution: revealing feedforward's true identity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 1931–1940. https://doi.org/10.1145/2470654.2466255

[38] 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

[39] Chen Zhu-Tian, Zeyu Xiong, Xiaoshuo Yao, and Elena Glassman. 2024. Sketch Then Generate: Providing Incremental User Feedback and Guiding LLM Code Generation through Language-Oriented Code Sketches. arXiv:2405.03998 [cs.HC] https://arxiv.org/abs/2405.03998