2024 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) | 979-8-3503-6613-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/VL/HCC60511.2024.00023

In Situ AI Prototyping: Infusing Multimodal Prompts into Mobile Settings with MobileMaker

Savvas Petridis*, Michael Xieyang Liu*, Alexander J. Fiannaca, Vivian Tsai, Michael Terry, Carrie J. Cai
Google DeepMind, USA
{petridis,lxieyang,afiannaca,vivtsai,michaelterry,cjcai}@google.com

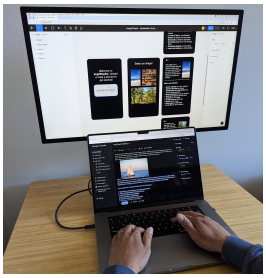
***Abstract*—Recent advances in multimodal large language models (LLMs) have made it easier to rapidly prototype AI-powered features, especially for mobile use cases. However, gathering early, mobile-situated user feedback on these AI prototypes remains challenging. The broad scope and flexibility of LLMs means that, for a given use-case-specific prototype, there is a crucial need to understand the wide range of in-the-wild input users are likely to provide and their in-context expectations for the AI's behavior. To explore the concept of *in situ* AI prototyping and testing, we created MobileMaker: a platform that enables designers to rapidly create and test mobile AI prototypes directly on devices. This tool also enables testers to make on-device, in-the-field revisions of prototypes using natural language. In an exploratory study with 16 participants, we explored how user feedback on prototypes created with MobileMaker compares to that of existing prototyping tools (e.g., Figma, prompt editors). Our findings suggest that MobileMaker prototypes enabled more serendipitous discovery of: model input edge cases, discrepancies between AI's and user's in-context interpretation of the task, and contextual signals missed by the AI. Furthermore, we learned that while the ability to make in-the-wild revisions led users to feel more fulfilled as active participants in the design process, it might also constrain their feedback to the subset of changes perceived as more actionable or implementable by the prototyping tool.**

***Index Terms*—Prototyping, LLMs, Generative AI, Design**

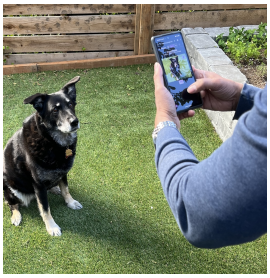
I. Introduction

Recent advances in LLMs have lowered the barriers to rapidly prototyping novel AI-powered features and products via prompt programming [1]–[4]. Compared to the traditional, more time-consuming AI development cycle of data collection, model training, and integration into UIs [5], [6], it is now much easier to prototype AI through "prompt-based prototyping" [7]–[9]: crafting LLM prompts in minutes and embedding them into new AI prototypes. Simultaneously, the multimodality of recent LLMs (e.g., Gemini [10], GPT4 [11]) has made it possible to process rich inputs (e.g., text, images, video), unlocking a wide range of use cases suitable for mobile or "on the go" settings, from diagnosing plant diseases (from a photo and text list of symptoms) to generating recipes (based on a picture of ingredients and set of dietary restrictions).

Despite rapid advancement of LLMs and AI prototyping, gathering *early tester feedback* on AI prototypes remains challenging, especially in mobile environments. Previous research on "embodied" user feedback [12]–[14] suggests that real-world, in-the-wild settings yield more authentic tester feedback than controlled lab settings, as prototypes are used within their intended context. This is especially crucial for LLM-powered AI prototypes due to LLMs' specific idiosyncrasies and flexibility. Unlike traditional machine learning models, which are typically scoped to specific, narrow applications with inputs and outputs relatively known a priori [15]–[17], LLMs can be applied to a broad spectrum of tasks (e.g., code completion, medical help, tutoring), receive diverse "natural" inputs (e.g., photography, icons, diagrams, text, audio), and function in various real-world settings (e.g., home, commute, work, coffee shop). Given this broad scope, for a given use case-specific prototype, there is the need to understand the types of *input* testers are likely to provide, their context-specific *expectations* of the AI's behavior, and validation of the overall prototype *concept*.



(a) Desktop prototyping and testing



(b) In situ prototyping and testing

Fig. 1: Testing AI prototypes on desktop, e.g. using UI mockups and LLM prompt editors (a) vs. on mobile (b): MobileMaker helps designers quickly get functional AI prototypes onto mobile devices and experienced in the wild for early feedback, and enables testers to revise and re-configure the AI prototype on-the-fly, while in the field.

LLMs also open up new opportunities for in situ prototyping and testing. First, because prompt programming has significantly reduced the ML expertise required to prototype AI, there is an opportunity for designers to more easily bring LLM-powered prototypes to life within mobile user interfaces (UIs) for in-the-wild experiences. Secondly, much as paper prototyping enables designers and testers[1] to alter designs on-the-fly, LLM-powered prototypes could enable testers to proactively revise and improve prototypes *during* testing, at the very moments they encounter an issue or get an idea for an improvement to the design.

To explore these opportunities for **in situ AI prototyping and testing**, we created MobileMaker: an AI prototyping

*Equal contribution.

[1]In this paper, we define "designers" as those who create prototypes, and "testers" as individuals who use these prototypes and provide user feedback.

979-8-3503-6613-6/24/$31.00 ©2024 IEEE
DOI 10.1109/VL/HCC60511.2024.00023

tool that 1) enables designers to rapidly create a mobile AI prototype that can be tested on-device, and 2) enables testers to revise the prototype's design in-the-wild simply through natural language (NL). Specifically, MobileMaker allows testers to provide not only feedback, but also alternative, functional prototypes that better meet their specific needs and can be instantly re-tested. We conducted an exploratory study to examine how tester feedback using MobileMaker (on functional LLM-powered prototypes) compares to and complements traditional user-feedback mechanisms, and to assess the impact of in-the-wild revisions on the testing experience. In a subsequent reflection activity, two professional designers discussed how the tester-generated revisions could influence the iterative design process.

We found that MobileMaker prototypes enabled testers to authentically experience the envisioned artifact, which allowed them to 1) evaluate if the overall concept was mobile appropriate, 2) experiment with serendipitous edge cases via unconventional, in-the-wild inputs, 3) discover discrepancies between their interpretation of the task and the AI's interpretation, and 4) more critically evaluate the AI's output using the richer contextual cues in their surroundings. We also discovered that the ability to make on-the-spot prototype revisions led testers to feel more engaged as *active participants* in the design process and more critically assess their own feedback. Furthermore, though designers were initially hesitant to let testers "short-circuit" the design process, they later recognized revised prototypes as a more powerful and convincing form of feedback, especially because they felt traditional user feedback could be more easily overlooked or dismissed. In sum, this paper makes the following contributions:

- Design goals for LLM-powered mobile AI prototyping platforms, based on formative studies and iterative feedback from designers,
- MobileMaker, a prototyping tool that enables designers to create, test, and revise AI prototypes in NL while on mobile,
- an exploratory study showing how in situ prototypes (created using MobileMaker) led to qualitatively different tester feedback compared to that of traditional AI prototypes,
- a follow-up reflection activity with two designers, showing how designers can make use of fully functional revised prototypes created by testers in the wild,
- a discussion of implications for future design and prototyping platforms, advocating for systems that prioritize quick iterations and authentic tester feedback for AI experienced in-the-wild.

II. Related Work

A. Prototyping with AI

Prototyping is a fundamental step that allows designers to explore and refine product ideas and user interfaces [18], [19]. Prototypes can range from low-fidelity [20], [21], which might just sketch out user flows [22], to high-fidelity, which closely mimic the final product in both look and functionality [23], [24]. However, integrating AI into these prototypes has been notoriously challenging–designers often struggle to grasp the capabilities and limitations of AI [25]–[28], and building functional prototypes of AI requires substantial ML expertise and engineering effort [29]–[31].

Recent advances in LLMs have dramatically reduced the barriers to prototyping AI functionality [1], [2], [32]–[34] through natural language prompting [3], [35]–[38]. In addition, systems like PromptInfuser [7], [8] and ProtoAI [39] have been developed to help designers prototype AI functionalities within the context of UI, enabling them to produce prototypes that realistically represent the envisioned artifact and better anticipating UI issues and technical constraints [8]. MobileMaker extends this capability by wrapping LLM-powered prompts and their outputs "in an app UI shell," with a particular focus on mobile scenarios. However, unlike previous approaches that focus on desktop-bound prototyping [8], [40], prototypes created with MobileMaker can be used on actual mobile devices, offering more authentic user experiences and interactions. Furthermore, MobileMaker supports transforming natural language ideas directly into working prototypes, reducing the initial barrier of building and configuring prototypes from scratch as well as the potential learning curve associated with onboarding to a new system.

B. In Situ User Testing

After developing a prototype, it is crucial to perform user testing and gather feedback [24], [41]. Traditional UI prototyping tools like Figma [42] allow testers to interact with click-through prototypes [42]–[44], but these often include only predefined input and output examples [20], [45], and the quality and relevance of these examples may not fully represent the diverse contexts and challenges that end-users may face in the real-world [46]. To circumvent the technical demands of building functional AI prototypes while still engaging end-users, designers often conduct user testing with Wizard of Oz setups [47], where a human simulates AI responses [48]–[50]. Although helpful, this often fails to accurately simulate the unique errors and behaviors of actual AI systems [51], [52].

To address these challenges, in this work, we focus on supporting "in situ" user testing of AI prototypes, i.e., enabling interactions with MobileMaker prototypes in their intended real-world environments [13], [14], [53], [54]. Specifically, MobileMaker allows users to capture inputs from their immediate surroundings using their mobile devices and view AI model outputs in that same context. We found that this approach not only facilitated the discovery of serendipitous edge cases from real-world inputs (echoing findings from [55], [56] in the context of in situ ML model testing), but also enabled testers to evaluate the AI's performance more critically, leveraging the rich contextual cues around them.

C. Iterative Design Based on User Feedback

In HCI, iterative design is recognized as a critical method that emphasizes the repeated cycle of designing, testing, and refining products based on user feedback [57]–[59], which is crucial in helping designers identify user pain points and areas

for improvements [58], [60]. The length of each iteration in the design process can vary significantly–ranging from a few days in rapid prototyping scenarios to several weeks or months in more sophisticated systems–based on factors like the iteration's specific objectives, the level of fidelity of the prototype involved, and the available resources [53], [61]–[63]. In this work, we explore the opportunity to accelerate these iteration cycles even further by having MobileMaker instantly implement users' feedback on the spot, thereby enabling them to immediately interact with a revised version of the prototype. This allowed our study participants to more critically reflect on their own feedback and increased their perceived engagement in the design process. Moreover, testers can submit these updated prototypes as "revisions" back to designers, providing a more direct means of conveying user needs in specific contexts.

III. Formative Studies & Design Goals

We began by conducting need-finding interviews with two professional UX designers (D1, D2) from a major technology company, both experienced in designing AI features. We explored their workflows for designing, prototyping, and testing LLM-powered features, as well as the challenges they faced. These discussions inspired the idea of rapidly creating LLM-powered mobile prototypes that can be immediately tested and revised in the wild, leading to the initial development of MobileMaker. Throughout the development process, we continued to solicit feedback from the original two designers as well as two additional designers (D3, D4), incorporating their insights and suggestions for improvement.

Based on the data from these formative studies as well as prior work, we identified three design goals for MobileMaker:

- **DG1: Rapid and frictionless authoring of medium-fidelity, mobile AI prototypes**. Designers typically prototype AI functionality and UI separately and with different tools. Integrating the AI and UI into a working prototype usually requires significant engineering resources and could often "take weeks if not months" (D2) with considerable coordination and communication overhead. While using early versions of MobileMaker, designers were excited about the possibility of being able to quickly wrap LLM-powered prompts "in an app shell" (D3) to create medium-fidelity prototypes. They noted that this would substantially accelerate prototyping velocity and reduce their reliance on engineers, as their goal was *not* to create high-fidelity, pixel-perfect prototypes or fully-fledged mobile applications. Three of them were also very interested in the ability to transform ideas into functional prototypes just by describing their ideas in NL.
- **DG2: Lowering the barriers to on-device testing experience and early-stage feedback**. Designers typically solicit early-stage feedback on AI prototypes by either showing testers static UI mockups, or by showing example model inputs and outputs via prompt editors. When presented with an early version of MobileMaker, designers found it compelling to potentially "get the best of both worlds" (D4)– a more dynamic and realistic approximation of the AI compared to UI mockups, and a more user-accessible UI compared to the raw model inputs and outputs seen in prompt editors. Furthermore, designers found it compelling to be able to go from an initial idea to a testable mobile interface quickly, as typical mobile development processes can be painstaking and slow. They were excited to have teammates and other users interact with their prototypes on-the-go, rather than being restricted to desktop environments.
- **DG3: Expedited design iteration loop**. Traditionally, refining a mobile prototype after user testing is a lengthy and resource-intensive process: designers must make revisions, collaborate with engineers to update the prototype, and schedule new testers, all of which can take days or weeks. To streamline this, designers envisioned a future where the feedback loop is drastically shortened. Designers were excited about the possibility of receiving tester feedback, then having an instant mechanism to produce an improved version of the prototype for another round of testing. This approach could not only keep testers more engaged and motivated compared to traditional methods, but also reduce designers' dependence on engineers for prototype updates, which typically could take "days or even weeks" (D1).

IV. MobileMaker

Based on the above design rationales, we describe the system and user interface design of MobileMaker. Our formative studies revealed that designers sought to make prototypes capable of unveiling user intentions, inputs, and reflections on model outputs without being overly complex. Thus, our objective was not to develop a platform for authoring fully-functional mobile applications; instead, we focused on implementing a minimal set of features essential for designers to explore the human-AI interactions of the prototypes.

A. Creating Prototypes

In MobileMaker, designers can build prototypes using three types of configurable UI widgets (Fig. 2):

- **Input Widgets** represent different modalities of input that designers can provide, including text, dropdown menus (with a list of predefined text values), live-camera photos (captured live from the front or back camera feed on the mobile device), and file uploads (e.g., photos from an on-device album) (Fig. 2A)
- **Action Widgets** represent user actions (e.g., a run button or timer) that trigger the prototype to run an LLM prompt with the user's input (Fig. 2B)
- **Output Widgets** are used to display the output from generative models, i.e., LLMs, multimodal LLMs, and image generation models (e.g., [64], [65]) (Fig. 2C).

In addition, output widgets have an editable prompt (Fig. $2D_1$) that is sent to the associated model when generating content. Prompts are composed of three text sections—model instructions, principles, and few-shot style examples–that are concatenated before being sent to a model. Prompts may reference the identifier for any input widget in order to have the user's input content from that widget merged into the prompt

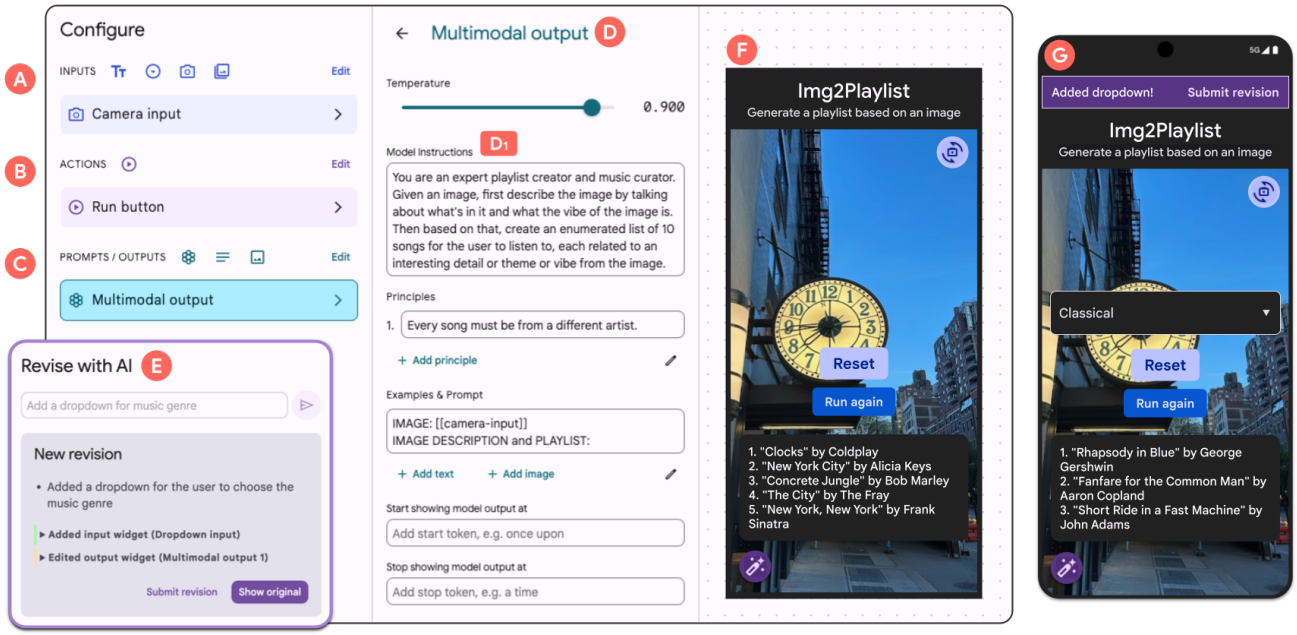


Fig. 2: **MobileMaker UI**. To build a prototype, users can add input (A), action (B), and output (C) widgets and customize their properties (D), e.g., editing an output widget's prompt (D1). Alternatively, users can create or revise prototypes with natural language via the "Revise with AI" panel (E). Changes are immediately rendered and can be tested in the mobile preview (F). Finally, users can test and revise the prototype on their phones in the wild (G).

when calling the model. Output widgets support configuration of model parameters (e.g., temperature) and output parsing behavior (e.g., stop tokens).

Each widget type offers customizable visual elements, such as text input *placeholders* (e.g., "Enter a color"), widget *labels* (e.g., "Upload a photo of your room"), and the choice to display model outputs as plain text or in a visual component (e.g., carousel card). Designers can also configure various global parameters, include the prototype's name, description, font style, and layout style, i.e., whether camera controls should be laid out in the list of input controls or as a full screen layout behind all other controls.

Under the hood, MobileMaker uses a JSON object to represent the set of widgets and configurations of a prototype (e.g., Fig. 3), which allows for easy remixing of prototypes (e.g., forking) and revising prototypes with LLMs (sec. IV-B2). MobileMaker renders JSON specifications into interactive prototypes by laying out UI controls for each configured widget (e.g., text input widgets appear as text boxes, photo input widgets appear as a camera interface with a live view from the phone's back camera; Fig. 2F). On a desktop browser, the prototype is rendered in a mobile phone preview viewport to the right of the configuration toolbars/panels (Fig. 2F); when viewed from a mobile device, the prototype is rendered as a full screen app, omitting the configuration toolbars/panels (Fig. 2G). Widget modifications are instantly visible from the preview, facilitating rapid iteration throughout the prototyping process.

B. Prototyping with Natural Language

From early feedback, we discovered that designers appreciate the capability to quickly convert ideas into functional mobile prototypes (**DG1**)–despite MobileMaker's straightforward interface, some designers found translating high-level concepts to specific low-level widgets cognitively demanding, and could hinder "spur of the moment" ideas (D3). To reduce this barrier, MobileMaker additionally supports prototyping through two NL features:

1) Creating with Natural Language: In order to allow designers to quickly bootstrap a prototype from NL, they can simply provide a brief sketch of their prototype idea (e.g., "a feature that generates a music playlist from a photo the user takes"). The system then uses a few-shot LLM prompt to instantly generate a prototype. Each example in the few-shot prompt consists of an input NL request and output JSON specification, which encompasses configurations for input, action, and output widgets and as detailed previously. Critically, output widget prompts are dynamically generated to reference the corresponding generated input widgets, reducing the need for further manual configuration.

2) Revising with Natural Language: To allow testers to rapidly modify prototypes (e.g., in response to in-the-wild inspirations), MobileMaker also supports NL revisions (Fig. 2E). To achieve this, they can simply provide a brief description of the desired changes in NL, such as "add a dropdown for music genre." The system then categorizes the request as either an update to an existing output widget prompt or an adjustment to the prototype structure (e.g., adding or removing widgets). For the former, the system executes a prompt-revision meta-prompt (see appendix A3) designed to modify an existing prompt based on the NL request. For the latter, the system combines the NL request and the current prototype JSON in a few-shot prompt

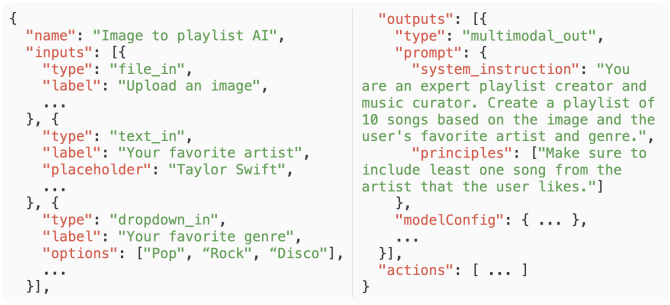


Fig. 3: Example prototype JSON representation.

(similar to the one for NL prototype creation, where each few-shot example includes a revision request, an initial JSON representation, and an updated JSON representation), which outputs a revised JSON specification for the entire prototype.

When someone requests an NL revision, they will receive a summary outlining the changes between the original prototype and the revised versions. They can toggle between these two versions and decide whether to accept the revision. This feature makes it easy for users to quickly compare and confirm if their requested revision has indeed been implemented (e.g., during in-the-wild testing), enhancing their ability to confidently and efficiently iterate on their design.

Together, these NL features enable rapid development and require minimal manual configuration, making MobileMaker's medium-fi prototyping process accessible and approachable to not only designers but also testers with limited prototyping or prompting experience.

C. Prototype Testing Experience on Mobile

Designers can share their prototype via URL, which enables team members or other testers to try it out in the wild (**DG2**). When running a prototype on a mobile device, each set of user inputs, model output, and user feedback is saved as a test case. Designers can review all collected test cases in a testing dashboard, allowing them to quickly explore use cases and discover unexpected edge cases.

1) Dynamic Prototype Revision: When testers experience unexpected outputs or prototype behavior, or they have feedback on how the prototype could be improved to better meet their needs, they can leverage the NL revision feature (sec. IV-B2) to generate a revised version of the prototype; this can be sent back to the prototype designer as a *suggested revision*. Each suggested revision includes the tester's revision request, the updated JSON prototype specification, and the latest test case generated with the revised prototype prior to submission. After revising, testers can choose to either continue testing with the modified prototype or revert to the original specification, allowing for both continued iteration on a specific concept and exploration of multiple orthogonal ideas.

2) Prototype Revision Dashboard: Prototype designers can use the revision dashboard (Fig. 4) to review and examine all proposed revisions, as well as dynamically try out each revision by applying its changes (Fig. 4A). When a revision is applied, MobileMaker renders the layout and functionality of its JSON prototype specification in the mobile phone

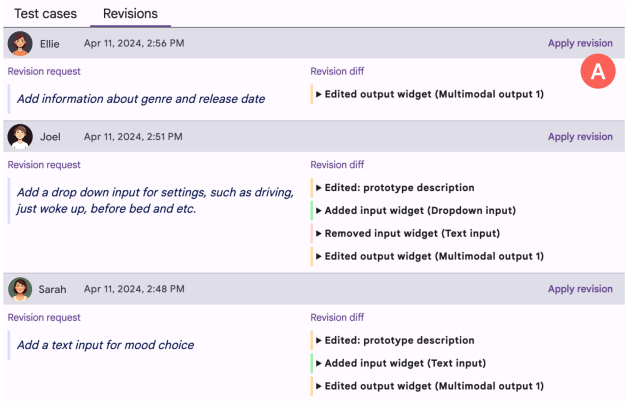


Fig. 4: Prototype Revision Dashboard

preview, then populates the preview with inputs and outputs from the associated test case that the original tester generated and experienced. This enables prototype designers to quickly understand and explore the impact of each revision on their prototype, significantly enhancing and accelerating the prototype design loop (**DG3**).

The feedback and revision features provide streamlined ways to capture testers' immediate reactions and ideas in the moment, creating a dataset of in situ responses for prototype creators to review during subsequent design iterations.

V. Exploratory User Study

Using MobileMaker as a probe, we conducted an exploratory study to understand: (1) how *in the wild* tester feedback on functional LLM-powered prototypes compares to status quo tester feedback collected on desktop and (2) how *revising* in the wild affects tester feedback.

The study was within-subjects with two conditions. In each condition, participants (henceforth referred to as "testers") were asked to give feedback on a prototype. In the MobileMaker condition, testers gave feedback on a MobileMaker prototype *in the wild*, and they were also able to *revise* the prototype with the NL revision feature. In the baseline (desktop) condition, testers clicked through a Figma prototype (with the same UI as the MobileMaker prototype) that demonstrated the basic interaction. Afterwards, they searched for images online to test the LLM prompt that would power the prototype in the Google AI Studio prompt editor. This baseline setup was informed by our formative studies with designers, who indicated that these are typical prototyping workflows their teams use to get early tester feedback on LLM-powered applications.

In the MobileMaker condition, testers used the NL revision feature to revise both the AI and UI functionality while on their phones. In the baseline condition, testers could alter the prompt in a text editor. In both conditions, if testers had trouble altering the prototype or if the NL revision did not work as expected (e.g., the revised prototype did not fit the tester's request), the study facilitators stepped in to help out.

The two prototypes testers experienced were both mobile, multimodal generative AI applications that two designers (D1&2) from our formative study ideated: (1) Img2Playlist, an application that generates a custom playlist based on

Measure	Statement (7-point Likert scale)
Ease	With {Setup A/B}, feedback came easily to me. I was easily stimulated to think of suggestions and feedback for the designer.
Communication	With {Setup A/B}, I felt like I could better communicate the changes I wanted to make to the prototype.
Actionable Feedback	With {Setup A/B}, I gave feedback that is easily actionable. I think designers can immediately apply my suggestions to their current application.
Enjoyable	With {Setup A/B}, it was enjoyable to interact with the prototype and provide feedback.
Authentic Experience	With {Setup A/B}, I felt like I had a realistic or authentic experience of the envisioned application.

TABLE I: **Post-task questionnaire** filled out by testers after they gave feedback on the prototypes from both conditions.

an input image, and (2) Img2VideoIdeas, an application that generates ideas for videos to create, also based on an input image. Each tester experienced both prototypes, one per condition and counterbalanced (e.g. P1 experienced Img2Playlist with MobileMaker and Img2VideoIdeas with the baseline, whereas P4 experienced Img2Playlist with the baseline and Img2VideoIdeas with MobileMaker). We decided on these applications because they target a general audience of users, have the same core interaction pattern, and are actual mobile, generative AI applications professional designers would like to build.

A. Procedure

The overall outline for the study is as follows: (1) Testers filled out a consent form prior to the study. (2) During the study, testers spent 50 minutes giving feedback on two different prototypes, one with MobileMaker (25 minutes) and the other with Figma and a prompt editor (25 minutes), all while thinking aloud. Condition-order and prototype-order were counterbalanced. (3) Testers then completed a post-study questionnaire that compared their experiences giving feedback on each prototype. (4) In a semi-structured interview, testers compared their experiences giving feedback in each condition. Each study session took one hour on average. To situate the task, in both conditions, testers were asked to imagine they were the target user of both of these mobile applications; give any and all feedback they had while interacting with both prototypes; and alter the prototype when they had an applicable revision in mind. In both conditions, if they felt they hit a dead end with their revisions, testers could restart from the beginning, either with the initial prototype (in the MobileMaker condition) or the initial prompt (in the baseline condition).

B. Participants

We recruited 16 participants (6 female, 10 male) from a large technology company, representing diverse professional backgrounds such as designers, product managers, software engineers, UX researchers. We did not target a specific background as the two test applications appeal to a broad audience. Participants were recruited via an email call for participation. Participants were recruited through an email invitation and participated in person in two metropolitan cities in New York City and Pennsylvania. During the study, they explored various

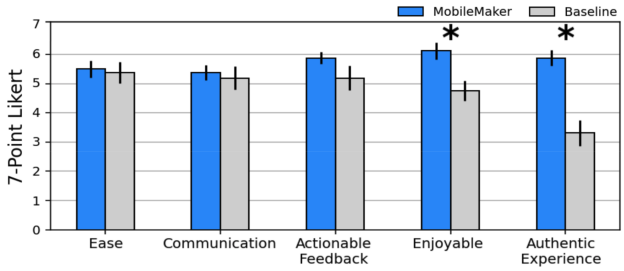


Fig. 5: Questionnaire results comparing the two conditions. Bars are standard error and an asterisk indicates a statistically significant difference (after full Bonferroni correction).

environments both within and outside office premises. Each participant received a $40 gift card for their participation.

C. Questionnaire

The questionnaire measures (Table I) are derived from established literature on qualities of good [66]–[68] and have been adapted to fit the perspective of testers. For instance, one quality of a prototype is its ability to capture the essence of the envisioned application [67]. We thus measured if testers felt they had authentically experienced the envisioned application. Finally, to compare the ratings from the two conditions, we conducted paired sample Wilcoxon tests with full Bonferroni correction, since the study was within-subjects and the questionnaire data was ordinal.

VI. Findings

Quantitative Findings. The results from the questionnaire are summarized in Figure 5. Testers reported that giving feedback on the MobileMaker prototypes was significantly more enjoyable (mean $= 6.12$, $\sigma = 1.11$) than giving feedback on the baseline prototype (mean $= 4.75$, $\sigma = 1.30$, $Z = 10$, $p = .01$), suggesting that walking about and experiencing the application in their local environment was a fun and novel experience for testers. In addition, testers also found MobileMaker provided a significantly more authentic experience (mean $= 5.88$, $\sigma = 1.05$) than the baseline prototype (mean $= 3.31$, $\sigma = 1.69$, $Z = 0$, $p < .01$), suggesting that testers felt that being able to experience the mobile prototype in the wild with real-world inputs provided a more realistic experience of the application. Finally, while MobileMaker was rated higher in ease, communication, and actionable feedback, the differences were not statistically significant. In the following sections, we provide further qualitative context to these results.

A. How Testing in the Wild Impacted Tester Feedback

Based on the questionnaire results, testers felt that MobileMaker provided a significantly more authentic experience than the baseline prototype, because they could interact with a functional prototype in the wild. Experiencing a functional prototype in the wild helped testers: (1) evaluate if the AI's output was mobile appropriate, (2) experiment with serendipitous edge cases via unconventional, in-the-wild inputs, (3) discover discrepancies between their interpretation of the task and the model's interpretation, and (4) more critically evaluate the model using contextual cues, leading

to their identification of rich contextual input in their surroundings that the model did not have access to. One limitation of in situ testing was that testers couldn't always easily get themselves into representative contexts (e.g., due to scheduling and location constraints), so it wasn't always possible to quickly find the most "canonical" inputs (e.g., for Img2Playlist, a coffee shop or driving commute picture).

1) MobileMaker helped testers evaluate if the product concept was mobile appropriate: While experiencing AI prototypes on MobileMaker, testers discovered ways in which the prototypes may not be compatible with in-the-wild contexts or form factors. For example, P4 revised Img2VideoIdeas to produce a video script along with each video idea, but after trying a few different image inputs, they soon realized that "generating a full script for the video ideas on the go is overwhelming and too much text." Beyond influencing considerations around form factor and mental capacity, the mobile vs. desktop context of use also affected user expectations around the overall "product fit" and appropriateness for mobile settings. For example, in the Img2VideoIdeas prototype, the model tended to output long-form content ideas, such as how-to videos and history-of videos. Though these may seem acceptable on desktop, P9 (who experienced Img2VideoIdeas in the MobileMaker condition) remarked that they expected more short-form, TikTok-style videos. In contrast, P16 (who experienced Img2VideoIdeas in the desktop condition) mentioned that the desktop setting may have prompted them to assume that the prototype was for creators making long-form content. Hence, by enabling testers to experience the prototype authentically on mobile, MobileMaker enabled testers to not only assess if the look and feel of the model output was mobile-appropriate, but also to reflect on the overall product concept.

2) Experiencing AI prototypes in the wild enabled serendipitous experimentation with edge cases: The photos that testers took in the wild tended to be "noisier" than on desktop, such as images taken at odd angles, images containing multiple objects of interest, images containing a combination of text and images, etc. The inherent noise in in-the-wild multimedia enabled testers to probe edge cases and boundaries of the model's capabilities and limitations. While testing Img2Playlist with MobileMaker, P1 noticed a picture of a dog on a nearby desk, with "Brandy" written at the edge. Curious to see if the model would pick up on this text, P1 input this photo, and to their surprise, the first song, "Brandy (You're a Fine Girl)" by Looking Glass, specifically referenced the text. P1 noted that, with MobileMaker, "you are walking around, which allows you to think more about ideas and edge cases," whereas on desktop, "I don't think I would have gotten that [an image with text] just sitting here [using image search]." Similarly, mobile constraints also led to situations in which the specific content intended to be captured was not always obvious to the model. For instance, P6 took a picture of a far-away cherry blossom, where the photo did not center on the blossom and included other objects in the scene. They were surprised when the model produced a general playlist about New York rather than the cherry blossoms. Overall, encountering these edge-case scenarios helped testers probe the boundaries of the model's capabilities and limitations.

3) Unconventional inputs uncovered discrepancies between the AI's and user's interpretation of the problem: In addition to encountering model edge cases (e.g., images with unique compositions), testers were also more likely to encounter objects and settings that were unique or unconventional given the *use case* (e.g., for Img2Playlist: pictures of hairspray, carton of milk, blank walls, etc.). This spontaneity and serendipity helped them uncover multiple possible interpretations of the problem. For example, while interacting with Img2Playlist on mobile, P2 input an image of a carton of milk (Figure 6b), expecting a playlist with a "coffee shop vibe," but instead received a more literal response: songs with titles related to milk (e.g., "Cream" by Prince). These spontaneous experiences uncovered multiple interpretations of the problem: it is unclear if a "relevant" playlist means that it ought to be relevant to the literal object in the picture, to the vibe of the scene, or to the look of an artist's album cover.

Meanwhile, on desktop, testers tended to search for conventional images that more often yielded expected model outputs. For example, P8 searched for "John Legend" in an image search engine and selected an image of one of his canonical album covers. They expected Img2Playlist to produce a playlist with songs only from that album, and it mostly did. This targeted approach lacked the serendipity of the MobileMaker inputs. As P8 explained, "*If I do a search on Google, I have an intention already... there's no randomness, and randomness is useful.*" It's possible that image search results may be less likely to expose intent mismatches between testers and the AI, because they themselves were found through being highly linked to a user's explicit text query. Overall, the images testers sourced in the baseline condition were more intent-driven and less ambiguous, which led to fewer discoveries of model output discrepancies.

4) Contextual cues enabled a more informed evaluation of the model's outputs: When experiencing prototypes on MobileMaker, testers had access to a wealth of sensory, spatial, and temporal context surrounding their image inputs, including the sounds and scents present in the environment, the encompassing landscape surrounding the image, and the events that occurred before and after the image was taken. This additional context helped them more critically evaluate model outputs that they would have otherwise deemed reasonable in the baseline condition. For example, P7 took a picture of two coworkers playing a relaxed game of pool in the early afternoon (Figure 6b), when most people had just finished eating lunch. They were surprised when the model erroneously identified the scene as "competitive." P7 realized that it may be difficult for a model to accurately assess the vibe with just a single, static image. "*On the phone, I was in the environment in which the photo was taken... I have more context, not just what's captured in that one-frame, but what's also before or after. It makes me more critical about the result.*" Similarly, P4 (while indoors) wished the model could take into consideration the fact that it was raining just outside their window, evoking

a solemn mood. Overall, testing the prototype in-situ not only enabled testers to assess the model's outputs through the surrounding context it was situated in, but also helped them identify contextual cues the model did *not* have access to.

B. How NL Revision Changed AI Prototyping Practices

Prompted by these in-the-wild discrepancies, testers revised the prototypes, ranging from *adding UI controls* (e.g. a dropdown menu for music genres or an input textbox to specify the audience for the video ideas) to *changing model behavior* (e.g. generating a description that associates each song in the playlist to the input image, or outputting a video script to accompany each idea). NL revision enabled testers to play an active, fulfilling role in the design process, and it also shortened the test-feedback loop. However, revising also potentially limited tester feedback to smaller, incremental revisions of the application's functionality. Whereas with Figma, testers gave higher level feature requests and ideas. This suggests that perhaps the feedback these prototypes provide are complementary and useful at different stages of the design process.

1) Revising enabled testers to provide better guardrails to the AI and reconsider the prototype's interaction pattern: When the model behavior did not match user intent, testers leveraged the NL revision tool to revise the prototype, such as by adding UI controls. As described in sec VI-A3, P2 discovered that the model sometimes generated playlists based on the literal object (e.g., "Cream" by Prince) rather than the vibe of the scene. When confronted with this issue, P8 revised their prototype to add a dropdown menu that allows testers to select whether the model should generate playlists based on the "vibe" or the "object" shown in the image. In subsequent testing, they found that having this input helped guardrail the model from solely providing playlists that were literally related to the objects in the image. Similarly, when P6 took a far-away picture of cherry blossoms, and the model did not pick up on the cherry blossom in the cluttered image (described in sec VI-A2), P6 added a textbox input widget so that testers can specify which object they wanted the playlist to be based on. After experimenting with this revision, P6 started to prefer having the playlist generated by the text. They appreciated the flexibility starting with text provided, and ultimately, they questioned the interaction pattern of starting with an image. Overall, the NL revision feature enabled testers to proactively react to model discrepancies, as well as critically reflect on the core interaction pattern (e.g., input and output types) of the application.

2) NL revise enabled testers to critically evaluate their own feedback in situ: With MobileMaker's NL revision feature, testers could immediately "live" and experience their own feedback within the updated prototype, which often helped them rethink their initial feedback. P4 initially revised Img2VideoIdeas to "Add a script for each video idea", but after testing the revised prototype, they realized that viewing a video script on mobile was overwhelming. They felt this realization could have only occurred on MobileMaker: "Walking around and trying it on mobile gave me a much better intuition on what I liked and disliked... for example, generating a full script for the video ideas on the go is overwhelming and too much text. I would have probably said I wanted that feature on desktop without realizing this. That realization is much more likely to happen if you're actually using it, as opposed to being asked to think about it." Beyond this, NL revision also empowered testers to make subsequent, "cascading" revisions (or cascading feedback based on a revision). In the example above, when P8 added the dropdown menu for selection of "vibe" or "object", they found that the model actually had difficulty assessing vibe from one image alone (described in sec VI-A4).

3) Revising helped shorten the test-feedback loop in MobileMaker: Revising in MobileMaker helped testers feel like they were actively and iteratively improving the prototype. While experiencing Img2Playlist, P2 added a dropdown menu for genre to control the types of songs output by the application and then added a description to accompany each song to explain their relevance to the input image. They felt that these additions tangibly improved the base application by introducing greater user control of the playlist and tightening the connection between the input image and output playlist: "In the mobile prototype, I felt like we were stepping up the stairs to get to a better and better, more refined version of the north-star version of the application." In the baseline, even though their feedback was used to alter the prompt in the editor, P2 felt they were giving feedback on the prompt's output, but the application as a whole was not improving. Echoing these thoughts, P16 described how experiencing their suggested revisions felt more active and fulfilling: "*You can complete this loop of giving feedback and seeing it in action. This experience is definitely much better, rather than describing your frustrations, and hopefully getting them fixed.*" Testers' revisions in MobileMaker would build on each other, which helped them feel (1) more active in the design process and (2) that they were tangibly improving the application.

4) Testers felt restricted to give feedback and revisions enabled by MobileMaker: While testers enjoyed immediately experiencing their feature suggestions, many also felt restricted by the revisions that were possible in MobileMaker. P6 explained, "*With [MobileMaker], you're a bit limited by what you can add to the UI. And whatever revision you add will be a bit more permanent in the mobile prototype.*" With MobileMaker, testers tended to focus on feedback that could lead to new revisions they could experience, e.g., adding a dropdown or a text input to steer the output playlist, potentially at the expense of providing feedback that could not be immediately rendered as a revision (e.g., thoughts on alternative user journeys or workflows). Meanwhile, in the baseline condition, testers often ideated more "out there" feature requests and entirely different functionalities with Figma. For example, P14 suggested that Img2VideoIdeas could be embedded into existing social media platforms and altered to also support caption generation for video posts. This higher level, more "out there" feedback elicited through Figma might be more useful to designers during early stage ideation, whereas the more granular and grounded feedback enabled by revising might be

(a) Example desktop inputs

(b) Example in-the-wild inputs

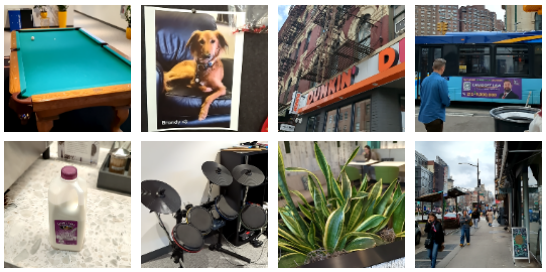


Fig. 6: Examples of test images tried by study participants during the desktop (a) and in-the-wild (b) conditions.

more useful further along the design process, when designers are refining their application's functionality. Overall, because they elicit quite different feedback, Figma and MobileMaker might be employed at different parts of the design process, depending on the needs of the designer.

5) Challenges in implementing revisions: While we intended for the testers to be able to independently update the prototypes, in practice, the facilitators occasionally needed to step in to help. In general, the LLM prompt powering revisions worked best when testers (1) input a single revision and (2) clearly specified that revision. For example, revision requests that added a single input like "add a dropdown for a music genre" or made a simple, clear adjustment to the model's output, e.g. "give me 5 songs instead of 10" tended to work well immediately. However, on a few occasions, testers entered revision requests with multiple changes (e.g., add two separate dropdowns and remove a text input) and the prompt failed to produce all of the specified changes. This type of error was mostly mitigated by educating testers on the feature during our short demonstration. One more challenging issue was clearly delineating requests. While P13 was testing Img2VideoIdeas, they wanted the application to take in multiple images that would be used together to inform three video ideas. They wrote: "I want three photos to be considered together," but the model interpreted this to mean creating a separate video idea for each image. If P13 had better specified their request, it likely would have been implemented correctly; however, testers often wanted to provide a quick description and have the model correctly extrapolate their intent.

VII. Design Reflection Exercise

In addition, we conducted an initial probe on how designers might interact with testers' prototype revisions. From the MobileMaker condition of the previous study, we presented a dashboard of participants' suggested revisions, as well as their revised prototypes, to the two designers who had initially prototyped Img2Playlist and Img2VideoIdeas. Designers spent around 30 minutes perusing the dashboard and applying the suggested revisions while thinking aloud.

D1 remarked that, relative to traditional forms of user feedback, experiencing users' dynamically-runnable revised prototypes was "very powerful and a more convincing form of feedback." They thought it made user feedback more tangible and compelling compared to vague requests such as "I want [feature request]," which are easier for designers to neglect or dismiss in practice. Beyond this, both designers found it crucial to have access to the *in-the-wild context* embedded within those revised prototypes. In MobileMaker, designers could run the revised prototype on the actual inputs experienced by the user (e.g., picture of pool table), and see the outputs the user saw. Those test cases also served as tangible evidence for suggested revisions. For example, the designers were initially perplexed by a sudden request to change the original Img2Playlist feature to "video2playlist." After noticing that the participant had previously inputted a series of photos depicting different scenes in a game room, they hypothesized that the participant preferred video over still images to better capture the dynamic context and ambiance. In the future, designers would like to see more of the "user journey" and reasoning behind the proposed revisions, so they could better understand users' motivations while running the prototypes.

Designers wished that there was an easier way to synthesize the different threads of feedback that participants provided in MobileMaker so that they could be more effectively utilized. One proposed solution by D2 was for MobileMaker to automatically organize the suggested revisions into distinct themes, such as "providing rationale for song choices in the output," "using video instead of image as input," or "add more UI controls to influence the type of playlist," as well as statistics on their frequency. As one designer explained, "it's not just one person reading through [the feedback] but a whole team that needs to be convinced that this needs to happen in this way," so it would be important to have sufficient evidence to help prioritize and facilitate more targeted design adjustments.

VIII. Discussion

A. Agile AI-Prototyping Practices in the Age of LLMs

Short design iteration loops coupled with frequent testing have long been advocated for in the design and software development communities. This paper expands on the question of what might be possible during the early phases of AI design if (1) designers can quickly push functional AI prototypes out into the wild and (2) testers can contribute both feedback and updated functional prototypes while in the field. In light of this, we may witness a parallel shift in the roles and responsibilities of designers, user researchers, developers, and end-users. Designers and developers might begin to borrow practices from user research (e.g., in-the-field user observations); likewise, user researchers may take on more designerly and creative roles by troubleshooting or refining user-proposed revisions directly in the field. Overall, in situ prototyping could introduce a blurring of boundaries between these traditional roles, broadening participation of AI development to include

more diverse professions and end-users. While this may inevitably introduce new frictions (e.g., increased coordination costs of disambiguating roles, potential suboptimal designs by end-users [61], etc.), future work could also capitalize on the new opportunities brought on by this blurring of roles, e.g., designers and developers more deeply considering off-the-desktop user needs; UX researchers taking a more active role in implementing user feedback; and end-user communities personalizing and configuring AI to their own needs.

B. Making In Situ Medium-fi Prototypes more "Sketchable"

Beyond giving passive feedback, MobileMaker users were empowered to actively participate in the prototyping process by revising prototypes and immediately "living" them in the wild. On one hand, this enabled users to more critically assess the feedback they gave (see sec VI-B2, where a participant suggested a revision, and realized after testing it that it was inappropriate for mobile). On the other hand, this introduced the potential risk that users might inadvertently "short circuit" the design process by prematurely discarding ideas that are not immediately feasible to implement within MobileMaker. Future work could address this by emphasizing to users that they are exploring initial feature ideas instead of final product implementation. One approach might involve modifying the UI of MobileMaker-built prototypes to resemble a sketch or low-fidelity paper prototype. Additionally, when running the NL revision feature, MobileMaker could produce multiple variations of the revision, e.g., three different video scripts varying in length and structure, to reinforce the concept that these are preliminary explorations with multiple potential implementations.

C. Uncovering Design Axes and Personas

While designers found the aggregated feedback and revisions useful for understanding user needs, they also expressed a desire to see the revisions clustered thematically so it would be easier to digest (see sec VII). Synthesizing this feedback at a larger scale, with perhaps a hundred users, could lead to a few interesting possibilities, such as uncovering key personas or revealing different conflicting perspectives on the future direction of the envisioned application. For example, some users of Img2Playlist might be "music experts" who want fine-grained controls to steer the playlist (e.g., through descriptors like "ethereal" or specific sub-genres); others might prefer a simpler UI with fewer controls so they can listen to music as soon as possible. These different perspectives could define a broad design space for Img2Playlist applications. In addition, future work could examine how LLMs might be leveraged to help designers transform aggregated feedback into a well-defined design space, both to better understand different user personas and product scope, and to identify under-explored regions within the design space. By uncovering early design axes, product scoping requirements, and personas for consideration, in situ prototyping could shift user feedback even further *upstream* in the AI development process.

D. Exploring Prototype Variants Based on Contexts

With the ability to instantly transform a piece of feedback or feature idea into an improved version of the prototype in situ, participants in the user study tended to engage in a "depth-first search" pattern, where they meticulously added or refined features through successive iterations. While thorough, this method risks leading testers down a narrow path, resulting in a form of tunnel vision where alternative ideas or unexpected user needs might be overlooked. To mitigate this, MobileMaker could encourage more breadth-first, "parallel-prototyping" [69] early in the testing process, such as leveraging the advanced reasoning capabilities of LLMs [35] to proactively suggest prototype variants based on both the existing prototype JSON configuration and the user's context (e.g., location, weather conditions). For a recommendation feature that suggests outfits based on a photo of the user's wardrobe and a selfie, MobileMaker could generate variants that let users provide additional input about their preferred style (e.g., casual, sporty, formal), or consider current weather and season. Beyond reducing design fixation, this approach might also encourage in-the-field testers to consider diverse conditions and assist them with prototyping AI that resonates with users' immediate and situational needs.

E. Limitations and Future Work

During the MobileMaker portion of the study, participants mostly explored areas inside or adjacent to their office buildings. To enrich results, participants could ideally use the applications in a variety of natural settings, such as near their homes or at favorite spots in the city. However, for practicality and to observe participants as they experienced the MobileMaker prototypes in person, we kept the locations standard across participants. Also, future work can examine how MobileMaker impacts tester feedback on a wider variety of applications. Finally, while MobileMaker's feature set is not exhaustive, it enabled us to probe the value and opportunities of in situ AI prototyping and testing in our exploratory study. In the future, MobileMaker can be expanded to support building prototypes with multi-screen, multi-LLM calls workflows and a wider array of UI inputs and outputs.

IX. Conclusion

To explore the opportunities for in situ AI prototyping and testing, we developed MobileMaker, with which designers can (1) rapidly create mobile AI prototypes and (2) enable testers to authentically experience the prototype in the wild and revise it in real-time. In our exploratory study, we found that by testing in the wild, testers were able to serendipitously experiment with image edge cases. In addition, NL revision enabled testers to critically evaluate their own feedback but also potentially limited the feedback they gave to what was actionable in MobileMaker. Future work might explore further supporting (1) users with providing feedback on in situ medium fidelity prototypes, as well as (2) designers with deriving insights from this feedback at scale.

References

[1] E. Jiang, K. Olson, E. Toh, A. Molina, A. Donsbach, M. Terry, and C. J. Cai, "PromptMaker: Prompt-based Prototyping with Large Language Models," in *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '22. New York, NY, USA: Association for Computing Machinery, Apr. 2022, pp. 1–8. [Online]. Available: https://dl.acm.org/doi/10.1145/3491101.3503564

[2] T. Wu, E. Jiang, A. Donsbach, J. Gray, A. Molina, M. Terry, and C. J. Cai, "PromptChainer: Chaining Large Language Model Prompts through Visual Programming," in *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '22. New York, NY, USA: Association for Computing Machinery, Apr. 2022, pp. 1–10. [Online]. Available: https://doi.org/10.1145/3491101.3519729

[3] T. Wu, M. Terry, and C. J. Cai, "AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts," in *CHI Conference on Human Factors in Computing Systems*, ser. CHI '22. New York, NY, USA: Association for Computing Machinery, Apr. 2022, pp. 1–22. [Online]. Available: https://doi.org/10.1145/3491102.3517582

[4] M. X. Liu, T. Wu, T. Chen, F. M. Li, A. Kittur, and B. A. Myers, "Selenite: Scaffolding Online Sensemaking with Comprehensive Overviews Elicited from Large Language Models," Oct. 2023. [Online]. Available: https://arxiv.org/abs/2310.02161

[5] Q. Yang, A. Scuito, J. Zimmerman, J. Forlizzi, and A. Steinfeld, "Investigating How Experienced UX Designers Effectively Work with Machine Learning," in *Proceedings of the 2018 Designing Interactive Systems Conference*, ser. DIS '18. New York, NY, USA: Association for Computing Machinery, Jun. 2018, pp. 585–596. [Online]. Available: https://dl.acm.org/doi/10.1145/3196709.3196730

[6] Q. Yang, A. Steinfeld, C. Rosé, and J. Zimmerman, "Re-examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–13. [Online]. Available: https://dl.acm.org/doi/10.1145/3313831.3376301

[7] S. Petridis, M. Terry, and C. J. Cai, "PromptInfuser: How Tightly Coupling AI and UI Design Impacts Designers' Workflows," Oct. 2023, arXiv:2310.15435 [cs]. [Online]. Available: http://arxiv.org/abs/2310.15435

[8] ——, "PromptInfuser: Bringing User Interface Mock-ups to Life with Large Language Models," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 1–6. [Online]. Available: https://dl.acm.org/doi/10.1145/3544549.3585628

[9] M. X. Liu, F. Liu, A. J. Fiannaca, T. Koo, L. Dixon, M. Terry, and C. J. Cai, ""We Need Structured Output": Towards User-centered Constraints on Large Language Model Output," Apr. 2024, arXiv:2404.07362 [cs]. [Online]. Available: http://arxiv.org/abs/2404.07362

[10] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, and others, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[11] OpenAI, "GPT-4 Technical Report," Mar. 2023, arXiv:2303.08774 [cs]. [Online]. Available: http://arxiv.org/abs/2303.08774

[12] P. Dourish, *Where the Action Is*. MIT Press, Aug. 2004. [Online]. Available: https://mitpress.mit.edu/9780262541787/where-the-action-is/

[13] Y. Rogers, K. Connelly, L. Tedesco, W. Hazlewood, A. Kurtz, R. E. Hall, J. Hursey, and T. Toscos, "Why It's Worth the Hassle: The Value of In-Situ Studies When Designing Ubicomp," in *UbiComp 2007: Ubiquitous Computing*, J. Krumm, G. D. Abowd, A. Seneviratne, and T. Strang, Eds. Berlin, Heidelberg: Springer, 2007, pp. 336–353.

[14] A. Crabtree, A. Chamberlain, R. E. Grinter, M. Jones, T. Rodden, and Y. Rogers, "Introduction to the Special Issue of "The Turn to The Wild"," *ACM Transactions on Computer-Human Interaction*, vol. 20, no. 3, pp. 13:1–13:4, Jul. 2013. [Online]. Available: https://dl.acm.org/doi/10.1145/2491500.2491501

[15] C. Harrison, D. Tan, and D. Morris, "Skinput: appropriating the body as an input surface," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: Association for Computing Machinery, Apr. 2010, pp. 453–462. [Online]. Available: https://dl.acm.org/doi/10.1145/1753326.1753394

[16] S. Hudson, J. Fogarty, C. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. Lee, and J. Yang, "Predicting human interruptibility with sensors: a Wizard of Oz feasibility study," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '03. New York, NY, USA: Association for Computing Machinery, Apr. 2003, pp. 257–264. [Online]. Available: https://dl.acm.org/doi/10.1145/642611.642657

[17] P. Langley, "Machine learning for adaptive user interfaces," in *KI-97: Advances in Artificial Intelligence*, G. Brewka, C. Habel, and B. Nebel, Eds. Berlin, Heidelberg: Springer, 1997, pp. 53–62.

[18] M. Beaudouin-Lafon and W. E. Mackay, "Prototyping Tools and Techniques," in *The Human-Computer Interaction Handbook*, 2nd ed. CRC Press, 2007.

[19] Y.-K. Lim, E. Stolterman, and J. Tenenberg, "The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas," *ACM Transactions on Computer-Human Interaction*, vol. 15, no. 2, pp. 7:1–7:27, Jul. 2008. [Online]. Available: https://doi.org/10.1145/1375761.1375762

[20] R. Sefelin, M. Tscheligi, and V. Giller, "Paper prototyping - what is it good for? a comparison of paper- and computer-based low-fidelity prototyping," in *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '03. New York, NY, USA: Association for Computing Machinery, Apr. 2003, pp. 778–779. [Online]. Available: https://doi.org/10.1145/765891.765986

[21] C. Snyder, *Paper prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann, 2003.

[22] M. De Sá and L. Carriço, "A mobile tool for in-situ prototyping," in *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. Bonn Germany: ACM, Sep. 2009, pp. 1–4. [Online]. Available: https://dl.acm.org/doi/10.1145/1613858.1613884

[23] J. Rudd, K. Stern, and S. Isensee, "Low vs. high-fidelity prototyping debate," *Interactions*, vol. 3, no. 1, pp. 76–85, Jan. 1996. [Online]. Available: https://dl.acm.org/doi/10.1145/223500.223514

[24] M. Walker, L. Takayama, and J. A. Landay, "High-Fidelity or Low-Fidelity, Paper or Computer? Choosing Attributes when Testing Web Prototypes," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 46, no. 5, pp. 661–665, Sep. 2002, publisher: SAGE Publications Inc. [Online]. Available: https://doi.org/10.1177/154193120204600513

[25] G. Dove, K. Halskov, J. Forlizzi, and J. Zimmerman, "UX Design Innovation: Challenges for Working with Machine Learning as a Design Material," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17. New York, NY, USA: Association for Computing Machinery, May 2017, pp. 278–288. [Online]. Available: https://doi.org/10.1145/3025453.3025739

[26] Q. Yang, J. Suh, N.-C. Chen, and G. Ramos, "Grounding Interactive Machine Learning Tool Design in How Non-Experts Actually Build Models," in *Proceedings of the 2018 Designing Interactive Systems Conference*, ser. DIS '18. New York, NY, USA: Association for Computing Machinery, Jun. 2018, pp. 573–584. [Online]. Available: https://doi.org/10.1145/3196709.3196729

[27] Q. Yang, N. Banovic, and J. Zimmerman, "Mapping Machine Learning Advances from HCI Research to Reveal Starting Places for Design Innovation," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. New York, NY, USA: Association for Computing Machinery, Apr. 2018, pp. 1–11. [Online]. Available: https://dl.acm.org/doi/10.1145/3173574.3173704

[28] M. X. Liu, A. Kittur, and B. A. Myers, "Crystalline: Lowering the Cost for Developers to Collect and Organize Information for Decision Making," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI '22. New York, NY, USA: Association for Computing Machinery, 2022, event-place: New Orleans, LA, USA. [Online]. Available: https://doi.org/10.1145/3491102.3501968

[29] F. Girardin and N. Lathia, "When user experience designers partner with data scientists," in *2017 AAAI Spring Symposium Series*, 2017.

[30] Q. Yang, J. Cranshaw, S. Amershi, S. T. Iqbal, and J. Teevan, "Sketching NLP: A Case Study of Exploring the Right Things To Design with Language Intelligence," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–12. [Online]. Available: https://doi.org/10.1145/3290605.3300415

[31] C. Kayacik, S. Chen, S. Noerly, J. Holbrook, A. Roberts, and D. Eck, "Identifying the Intersections: User Experience + Research

Scientist Collaboration in a Generative Machine Learning Interface," in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '19. New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–8. [Online]. Available: https://dl.acm.org/doi/10.1145/3290607.3299059

[32] S. Petridis, B. Wedin, J. Wexler, A. Donsbach, M. Pushkarna, N. Goyal, C. J. Cai, and M. Terry, "ConstitutionMaker: Interactively Critiquing Large Language Models by Converting Feedback into Principles," Oct. 2023, arXiv:2310.15428 [cs]. [Online]. Available: http://arxiv.org/abs/2310.15428

[33] M. X. Liu, A. Sarkar, C. Negreanu, B. Zorn, J. Williams, N. Toronto, and A. D. Gordon, ""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*, ser. CHI '23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 1–31. [Online]. Available: https://dl.acm.org/doi/10.1145/3544548.3580817

[34] J. J. Y. Chung, W. Kim, K. M. Yoo, H. Lee, E. Adar, and M. Chang, "TaleBrush: Sketching Stories with Generative Pretrained Language Models," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI '22. New York, NY, USA: Association for Computing Machinery, Apr. 2022, pp. 1–19. [Online]. Available: https://doi.org/10.1145/3491102.3501819

[35] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," Jul. 2020, arXiv:2005.14165 [cs]. [Online]. Available: http://arxiv.org/abs/2005.14165

[36] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," Mar. 2022, arXiv:2203.02155 [cs]. [Online]. Available: http://arxiv.org/abs/2203.02155

[37] M. Kahng, I. Tenney, M. Pushkarna, M. X. Liu, J. Wexler, E. Reif, K. Kallarackal, M. Chang, M. Terry, and L. Dixon, "LLM Comparator: Visual Analytics for Side-by-Side Evaluation of Large Language Models," Feb. 2024, arXiv:2402.10524 [cs]. [Online]. Available: http://arxiv.org/abs/2402.10524

[38] S. Petridis, B. Wedin, A. Yuan, J. Wexler, and N. Thain, "ConstitutionalExperts: Training a Mixture of Principle-based Prompts," Mar. 2024, arXiv:2403.04894 [cs]. [Online]. Available: http://arxiv.org/abs/2403.04894

[39] H. Subramonyam, C. Seifert, and E. Adar, "ProtoAI: Model-Informed Prototyping for AI-Powered Interfaces," in *Proceedings of the 26th International Conference on Intelligent User Interfaces*, ser. IUI '21. New York, NY, USA: Association for Computing Machinery, Apr. 2021, pp. 48–58. [Online]. Available: https://doi.org/10.1145/3397481.3450640

[40] K. J. K. Feng, Q. V. Liao, Z. Xiao, J. W. Vaughan, A. X. Zhang, and D. W. McDonald, "Canvil: Designerly Adaptation for LLM-Powered User Experiences," Jan. 2024. [Online]. Available: http://arxiv.org/abs/2401.09051

[41] C. Boothe, L. Strawderman, and E. Hosea, "The effects of prototype medium on usability testing," *Applied Ergonomics*, vol. 44, no. 6, pp. 1033–1038, Nov. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003687013000860

[42] Figma, "Figma: The Collaborative Interface Design Tool," 2024. [Online]. Available: https://www.figma.com/

[43] Balsamiq, "Balsamiq: Fast, focused wireframing for teams and individuals | Balsamiq," 2024.

[44] Sketch, "Sketch," 2024. [Online]. Available: https://www.sketch.com/

[45] K. J. K. Feng and D. W. Mcdonald, "Addressing UX Practitioners' Challenges in Designing ML Applications: an Interactive Machine Learning Approach," in *Proceedings of the 28th International Conference on Intelligent User Interfaces*, ser. IUI '23. New York, NY, USA: Association for Computing Machinery, Mar. 2023, pp. 337–352. [Online]. Available: https://dl.acm.org/doi/10.1145/3581641.3584064

[46] Q. Yang, "Machine Learning as a UX Design Material: How Can We Imagine Beyond Automation, Recommenders, and Reminders?" *AAAI Spring Symposia*, vol. 1, no. 2.1, pp. 2–6, Mar. 2018.

[47] D. Maulsby, S. Greenberg, and R. Mander, "Prototyping an intelligent agent through Wizard of Oz," in *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, ser. CHI '93. New York, NY, USA: Association for Computing Machinery, May 1993, pp. 277–284. [Online]. Available: https://dl.acm.org/doi/10.1145/169059.169215

[48] J. Cranshaw, E. Elwany, T. Newman, R. Kocielnik, B. Yu, S. Soni, J. Teevan, and A. Monroy-Hernández, "Calendar.Help: Designing a Workflow-Based Scheduling Agent with Humans in the Loop," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17. New York, NY, USA: ACM, 2017, pp. 2382–2393. [Online]. Available: http://doi.acm.org/10.1145/3025453.3025780

[49] S. R. Klemmer, A. K. Sinha, J. Chen, J. A. Landay, N. Aboobaker, and A. Wang, "Suede: a Wizard of Oz prototyping tool for speech user interfaces," in *Proceedings of the 13th annual ACM symposium on User interface software and technology*, ser. UIST '00. New York, NY, USA: Association for Computing Machinery, Nov. 2000, pp. 1–10. [Online]. Available: https://dl.acm.org/doi/10.1145/354401.354406

[50] L. D. Riek, "Wizard of Oz studies in HRI: a systematic review and new reporting guidelines," *Journal of Human-Robot Interaction*, vol. 1, no. 1, pp. 119–136, Jul. 2012. [Online]. Available: https://dl.acm.org/doi/10.5898/JHRI.1.1.Riek

[51] C. Parnin, G. Soares, R. Pandita, S. Gulwani, J. Rich, and A. Z. Henley, "Building Your Own Product Copilot: Challenges, Opportunities, and Needs," Dec. 2023, arXiv:2312.14231 [cs]. [Online]. Available: http://arxiv.org/abs/2312.14231

[52] C. Kulkarni, S. Druga, M. Chang, A. Fiannaca, C. Cai, and M. Terry, "A Word is Worth a Thousand Pictures: Prompts as AI Design Material," Mar. 2023, arXiv:2303.12647 [cs]. [Online]. Available: http://arxiv.org/abs/2303.12647

[53] J. C. d. A. Nogueira, A. S. Gomes, A. S. d. C. Filho, and F. Moreira, "Effectiveness of embodied evaluation of mobile applications: A qualitative study," *Heliyon*, vol. 9, no. 6, p. e17043, Jun. 2023.

[54] D. Dzvonyar, S. Krusche, R. Alkadhi, and B. Bruegge, "Context-Aware User Feedback in Continuous Software Evolution," in *2016 IEEE/ACM International Workshop on Continuous Software Evolution and Delivery (CSED)*, May 2016, pp. 12–18.

[55] T. Tseng, M. J. Davidson, L. Morales-Navarro, J. K. Chen, V. Delaney, M. Leibowitz, J. Beason, and R. B. Shapiro, "Co-ML: Collaborative Machine Learning Model Building for Developing Dataset Design Practices," *ACM Trans. Comput. Educ.*, vol. 24, no. 2, pp. 25:1–25:37, Apr. 2024. [Online]. Available: https://doi.org/10.1145/3641552

[56] U. Dwivedi, J. Gandhi, R. Parikh, M. Coenraad, E. Bonsignore, and H. Kacorri, "Exploring Machine Teaching with Children," Sep. 2021, arXiv:2109.11434 [cs]. [Online]. Available: http://arxiv.org/abs/2109.11434

[57] W. Buxton and R. Sniderman, "Iteration in the design of the human-computer interface," in *Proc. of the 13th annual meeting, Human Factors Association of Canada*, 1980, pp. 72–81.

[58] S. Dow, B. MacIntyre, J. Lee, C. Oezbek, J. Bolter, and M. Gandy, "Wizard of Oz support throughout an iterative design process," *IEEE Pervasive Computing*, vol. 4, no. 4, pp. 18–26, 2005.

[59] J. Nielsen, "Iterative Design of User Interfaces," 1993. [Online]. Available: https://www.nngroup.com/articles/iterative-design/

[60] J. Knapp, J. Zeratsky, and B. Kowitz, *Sprint: How to solve big problems and test new ideas in just five days*. Simon and Schuster, 2016.

[61] D. Norman, *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[62] K. T. Ulrich and S. D. Eppinger, *Product design and development*. McGraw-hill, 2016.

[63] T. Brown and B. Katz, "Change by design," *Journal of product innovation management*, vol. 28, no. 3, pp. 381–383, 2011, publisher: Wiley Online Library.

[64] "Overview of multimodal models," publication Title: Google Cloud. [Online]. Available: https://cloud.google.com/vertex-ai/generative-ai/docs/multimodal/overview

[65] "Imagen on Vertex AI AI Image Generator," publication Title: Google Cloud. [Online]. Available: https://cloud.google.com/vertex-ai/generative-ai/docs/image/overview

[66] M. Beaudouin-Lafon and W. E. Mackay, "Prototyping tools and techniques," in *The human-computer interaction handbook*. CRC Press, 2007, pp. 1043–1066.

[67] Y.-K. Lim, E. Stolterman, and J. Tenenberg, "The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas," *ACM Trans. Comput.-Hum. Interact.*, vol. 15, no. 2, jul 2008. [Online]. Available: https://doi.org/10.1145/1375761.1375762

[68] D. A. Schön, *The reflective practitioner: How professionals think in action*. Routledge, 2017.

[69] S. P. Dow, A. Glassco, J. Kass, M. Schwarz, D. L. Schwartz, and S. R. Klemmer, "Parallel prototyping leads to better design results, more divergence, and increased self-efficacy," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 17, no. 4, pp. 1–24, 2010, publisher: ACM New York, NY, USA.