Int. J. Human–Computer Studies 206 (2025) 103661

Contents lists available at ScienceDirect

International Journal of Human - Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Toward AI-driven UI transition intuitiveness inspection for smartphone apps

Xiaozhu Hu [a], Xiaoyu Mo [b], Xiaofu Jin [b], Yuan Chai [c], Yongquan Hu [d], Mingming Fan [e], Tristan Braud [a],*

[a] *Division of Integrative Systems and Design, The Hong Kong University of Science and Technology, Hong Kong, China*
[b] *IIP (Computational Media and Arts), The Hong Kong University of Science and Technology, Hong Kong, China*
[c] *Innovation, Policy and Entrepreneurship Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong, China*
[d] *National University of Singapore, Singapore*
[e] *Computational Media and Arts Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong, China*

ARTICLE INFO

Keywords:
UI transition intuitiveness
AI-driven user simulation
Design inspection

ABSTRACT

Participant-involved formative evaluations is necessary to ensure the intuitiveness of UI transition in mobile apps, but they are neither scalable nor immediate. Recent advances in AI-driven user simulation show promise, but they have not specifically targeted this scenario. This work introduces UTP (UI Transition Predictor), a tool designed to facilitate formative evaluations of UI transitions through two key user simulation models: 1. Predicting and explaining potential user uncertainty during navigation. 2. Predicting the UI element users would most likely select to transition between screens and explaining the corresponding reasons. These models are built on a human-annotated dataset of UI transitions, comprising 140 UI screen pairs and encompassing both high-fidelity and low-fidelity counterparts of UI screen pairs. Technical evaluation indicates that the models outperform GPT-4o in predicting user uncertainty and achieve comparable performance in predicting users' selection of UI elements for transitions using a lighter, open-weight model. The tool has been validated to support the rapid screening of design flaws, and the confirmation of UI transitions appears to be intuitive.

1. Introduction

Designing user-friendly mobile interfaces is crucial for the success of mobile applications. An essential part is ensuring that first-time users can confidently navigate between UI screens by selecting the correct "link UI elements" (i.e., buttons or links that trigger transitions) based on the current screen content and their mental abstraction of the target screen (Li and Luximon, 2023; Dørum and Garland, 2011). Designers often rely on familiar navigation patterns and design principles to create intuitive UI transitions but they may not generalize to all apps and can sometimes conflict with user expectations. Therefore, formative evaluations such as cognitive walkthroughs and quick usability tests with users are necessary during design iterations, from low-fidelity UI sketches to connected high-fidelity screens.

In UX design, running formative evaluations with users requires significant resources, limiting scalability and delaying feedback. On the other hand, designers' expertise can bias judgment when role-playing as users. Motivated by evidence that large language models (LLMs) can mimic human judgments from textual descriptions (Kosinski, 2023), there has been a growing interest in leveraging them to generate usability insights. Prior systems such as UXAgent (Lu et al., 2025) and SimUser (Xiang et al., 2024) employ LLMs to simulate usability feedback but assume highly complete, executable apps, and target large-scale usability testing. How to facilitate the formative evaluation of UI transitions by simulating user interactions based on early-stage, non-executable UI prototypes remains to be explored. Additionally, advances in UI understanding and agentic control have enabled agents that automate UI tasks (Hong et al., 2024; Rawles et al., 2024; Yang et al., 2023; Wang et al., 2024; Yan et al., 2023), but these agents are commonly trained on app-harvested traces (Rawles et al., 2024; Deka et al., 2017) or LLM-generated data (Jiang et al., 2023) and optimized to finish tasks. As a result, their action traces can diverge from first-time users' behavior and cognition, a gap exacerbated by the lack of datasets annotated with first-time users' UI transition choices.

This work focuses on facilitating formative evaluations of UI transitions throughout the design process by simulating users' real UI transition behaviors. We develop the UI Transition Predictor (UTP), an AI-driven tool that provides instant user simulation support for designers from low-fidelity UI mockups to high-fidelity UI screens. UTP targets two primary concerns of designers during formative evaluations of UI transitions: the confidence of the user in choosing the link UI element as well as the UI element they will most likely pick (Candiasa

* Corresponding author.
E-mail address: braudt@ust.hk (T. Braud).

https://doi.org/10.1016/j.ijhcs.2025.103661
Received 20 May 2025; Received in revised form 16 October 2025; Accepted 29 October 2025
Available online 1 November 2025

et al., 2023; Cepeda et al., 2021), addressed through the two following functions: 1. **Uncertainty prediction**: Predicting and explaining how uncertain users might feel on selecting the correct link UI element to the target screen. 2. **Link UI prediction**: Predicting the link UI element users would most likely select to transition between screens and explaining the reason behind the prediction. The uncertainty prediction results help ensure that users select link UI elements with confidence, rather than guessing among several similar options. The link UI prediction further helps designers assess whether the link UI element selected by users is consistent with their expected design decision.

To enable faithful user simulation functions of the UTP, we first conduct a data collection study with 40 participants to create a validated human-annotated benchmark dataset of UI transitions. The dataset consists of 140 pairs of UI screens. Each pair is annotated by 20 participants, comprising their first-time choices of link UI elements and the reasons for such choices. Our dataset includes low-fidelity and high-fidelity designs of each pair of UI screens, allowing assessment from the early stages of the design process to the final prototypes. We also collect participants' common strategies for identifying link UI elements and sources of uncertainty, which inform our model design and strategies to prompt LLMs.

We apply the dataset to develop user simulation functions of UTP. GPT-4o is adopted as the baseline for all user simulation functions due to its state-of-the-art performance across various domains (Lu et al., 2024), integrating different prompting strategies and utilizing the human-annotated data. Due to the limited transparency of commercial models and the ever-changing nature of their training data, we also train a binary classification model for uncertainty prediction and fine-tune an open-source, memory-efficient LLM (LLaMa-3.1-8B) for link UI prediction. Our uncertainty prediction model consistently outperforms GPT-4o in identifying UI transitions that cause user uncertainty during navigation, both in low- and high-fidelity prototypes. Similarly, our fine-tuned LLaMa-3.1-8B model achieves comparable accuracy to well-prompted GPT-4o in predicting user-anticipated link UI elements (64.35% vs. 71.30%).

We implement these user simulation functions into two interaction modes to support different scenarios: 1. **Screening mode**: An automated mode of UTP, which allow designers to quickly screen potential design flaws; 2. **Dedicated mode**: A semi-automated mode where designers can manually mark candidate UI elements that could be confusing to users. UTP disambiguates and decides which one the user is likely to pick.

We evaluated the effectiveness of the user simulation functions of UTP for designers through a controlled experiment involving 12 UI/UX practitioners. Participants were asked to inspect UI transitions both with and without the assistance of UTP. To ensure that any observed human performance or confidence stems from the presentation and use of accurate simulation signals, rather than being confounded by model prediction errors, we used only verified correct predictions from UTP's simulation results in the study. UTP's screening mode significantly improves the designers' accuracy of inspecting UI transitions to 75% (58.33% for without UTP and 62.50% for dedicated mode) while reducing the time cost by 65.70% compared to the condition without UTP, and by 59.43% compared to the dedicated mode of UTP. Our study further reveals a clear division of strengths between UTP's two modes. The screening mode effectively improves detection accuracy on unintuitive UI transitions to 75.00% compared to 41.67% without UTP and 33.33% with the dedicated mode. The dedicated mode should not be used as a stand-alone inspection method due to the low accuracy for detecting unintuitive UI transitions. However, it effectively helps with verifying UI transitions that already appear intuitive, achieving 91.67% accuracy versus 75.00% for both screening and manual inspection. Therefore, UTP can be adopted through a staged design workflow: screening to surface potential issues first, then confirming UI flows that are likely intuitive. Additionally, participants also report significantly higher confidence when utilizing the screening mode (Median = 5/5, IQR = 1.25) compared to the inspection without UTP (Median = 4/5, IQR = 1.5) and with dedicated mode (Median = 4/5, IQR = 1).

The contribution of this work can be summarized as follows.

1. A human-annotated benchmark dataset of UI transitions and the corresponding human factor insights to support the development of AI models for user simulation on UI transitions.
2. AI models and prompting strategies built on the human-annotated dataset, to predict and explain users' selection of link UI elements.

2. Related work

We examine previous research under four aspects: design of GUI navigation, design evaluation methods, UI evaluation with user simulation methods, and UI understanding with LLM.

2.1. GUI navigation design

The inception of the Graphical User Interface (GUI) introduced UI components for linking separate interfaces, facilitating users to navigate either to a new page or another section within the current page (Tidwell, 2010). We segment the vast body of research on GUI navigation design into macro and micro levels. Research on GUI navigation which was conducted from a macro perspective, emphasize design guidelines (Kalbach, 2007; Nielsen, 1999; Farkas and Farkas, 2000), conceptual models (Webster and Ahuja, 2006; Djonov, 2007), and interactive techniques (Baeg et al., 1994) to rationalize the hierarchical structure of web information and simplify users' information retrieval. Conversely, micro-level insights into GUI navigation design focus on the logic of UI transition between two consecutive GUI screens. Chang and Dillon (1998) propose that intuitive UI navigation includes the logical presentation of static UI elements and the logical way to chain the preconditions and postconditions of each navigational event. To this end, semantic-based design concepts like interface metaphors are commonly applied to facilitate users' understanding of UI transition logic (Lakoff and Johnson, 2008; Barr et al., 2002), and laws of visual perception are playing a role in directing the consumer flow (Koch and Oulasvirta, 2016). Extensive design guidelines exist for crafting UI elements to be visually attractive and semantically clear, particularly within typical web (Farkas and Farkas, 2000; Burrell and Sodan, 2006) and mobile applications (Punchoojit et al., 2017). Although existing design guidelines are useful for facilitating users to perceive and learn the UI semantics quickly, research on how users understand the transition logic between consecutive UI screens is lacking.

2.2. Design evaluation methods

Typical design evaluation methods can be classified into user-participated and designer-centered approaches (Granollers and Lorés, 2004). User-participated methods typically involve testing with representative users who perform specific tasks using a given prototype, such as usability tests (Arning and Ziefle, 2009; Leesutthipornchai and Pradubsuwun, 2023) and A/B tests (Vanderdonckt et al., 2019). Designer-centered methods, also known as design inspection methods (Nielsen, 1994; Granollers and Lorés, 2004), include techniques like heuristic evaluation (Nielsen, 1992; Nielsen and Molich, 1990) and cognitive walkthrough (Polson et al., 1992; Lewis and Wharton, 1997). Due to practical constraints such as time and budget, user-participated evaluations cannot be frequently or immediately conducted during design iterations. As a result, designers often rely on designer-centered methods, which are more cost-effective and can be applied at various stages of prototyping (Nielsen, 1994). For example, heuristic evaluation is a widely used designer-centered method that employs a set of guidelines to identify and characterize undesirable interface features (Nielsen, 1992; Nielsen and Molich, 1990). However, this method reflects feedback from the designer's perspective, lacking direct insights from users. Cognitive walkthroughs provide a more user-centered method, where designers act as the users, walking through the UIs step by step to predict potential user feedback (Polson et al., 1992; Lewis and Wharton, 1997; Nielsen, 1994). However, since real users are not involved, these evaluations often miss important issues that typical users might encounter, primarily due to the differences in how design experts and regular users understand the UI (Granollers and Lorés, 2004; Mahatody et al., 2010). Effectively and instantly accessing users' feedback in the design inspection is valuable but challenging for designers.

2.3. UI evaluation with user simulation

Evaluating UI design through user testing is often time-consuming, labor-intensive, and costly. To address this, AI-driven systems have emerged to help designers instantly evaluate their UI prototypes. Some techniques utilize rule-based pipelines to automatically detect whether a UI prototype adheres to general design guidelines (Yang et al., 2021; Chen et al., 2017; Blackmon et al., 2007). Others follow a data-driven approach to evaluate UI designs by predicting how users may perceive and interact with the interface (Wu et al., 2020; Schoop et al., 2022; Bellamy et al., 2011; Kim et al., 2022; Wu et al., 2019; Bylinskii et al., 2017; Swearngin and Li, 2019; Fosco et al., 2020; Lee et al., 2020). These methods also predict specific UX metrics, such as task completion time (Bellamy et al., 2011; Biswas and Robinson, 2010), user engagement (Wu et al., 2020), brand personality (Wu et al., 2019), and the tappability of GUI elements (Swearngin and Li, 2019; Schoop et al., 2022). With large language models (LLMs) demonstrating the ability to mimic human characteristics through text (Kosinski, 2023), recent research has explored their potential for UI evaluation. Duan et al. (2024) leverage GPT-4 to automatically generate expert feedback on UI mockups for heuristic evaluation, while Simuser (Xiang et al., 2024) and UXAgent (Lu et al., 2025) both introduce multi-agent systems to facilitate an LLM to simulate various users' perception of UI contents, generate user flows on interactive prototypes, and provide with usability feedback on whether the UI content aligns with user expectations. However, current research overlooks a common cognitive process users engage in when navigating between UI screens: identifying the most likely UI element to reach the desired UI screen (link UI element) from multiple options on the current UI screen. Simulating this process is important because users' selection of link UI elements, whether correct or incorrect, informs the intuitiveness of the mapping between UI elements on one UI screen and their corresponding target UI screens. Ensuring such intuitive UI transitions is essential to facilitating smooth user flows toward desired screens. In this work, we aim to address this gap by focusing on two primary prediction tasks: predicting user uncertainty about which UI element will link to the desired screen and identifying the most likely UI element users will select to navigate to the target screen. We also investigate methods for generating explanations for these predictions.

2.4. UI understanding with LLM

Enabling AI models to understand UI content and support the workflows of users or UI designers has been a long-standing goal in HCI research. Before the advent of LLM, it was common practice to train task-specific models using large datasets for UI-related tasks such as tappability prediction (Swearngin and Li, 2019; Schoop et al., 2022), UI screen summarization (Wang et al., 2021), UI element detection and metadata reconstruction (Li et al., 2020; Zhang et al., 2021; Chen et al., 2022; Wu et al., 2021), UI element relationship recognition (Wu et al., 2023), UI screen relationship recognition (Feiz et al., 2022), and predicting the UI element linking two UI screens (He et al., 2021; Johns et al., 2023). With the rise of textual and multimodal LLMs, researchers have begun exploring how to train, fine-tune, and prompt large foundational models to improve their performance across multiple UI tasks. For instance, Spotlight (Li and Li, 2022) presents a vision-based approach for a range of mobile UI understanding tasks, while Wang et al. (2023) propose representing mobile UI screens as HTML to enhance GPT-2's performance on several UI tasks. As multimodal LLM capabilities grow, there has been a surge of work on collecting specialized datasets (Jiang et al., 2023; Rawles et al., 2024), fine-tuning open-source models (Hong et al., 2024; Si et al., 2024), and prompting closed-source models like GPT-4V in automated systems such as AppAgent (Yang et al., 2023), Mobile-Agent (Wang et al., 2024), and MM-Navigator (Yan et al., 2023), laying the groundwork for automatic GUI agents. However, there remains a significant research gap in using advanced LLMs to simulate user interactions. There is a need for systematic exploration of how textual or multimodal LLMs can simulate users' selection of UI elements during transitions and their understanding of the logic behind these transitions. The lack of datasets is a key obstacle. Although the Rico$_{\text{link}}$ (Deka et al., 2017; Johns et al., 2023) and AITW (Rawles et al., 2024) projects provide datasets on UI transitions, and IluvUI (Jiang et al., 2023) uses large language models (LLMs) to create vision-language datasets in the UI domain, a significant challenge remains. There is no dataset annotated by humans that captures the real users' anticipated link UI elements, the uncertainty in the selection process, and the reasons behind these choices. To address this gap, we have contributed a human-annotated benchmark dataset. Additionally, recognizing that the logic of UI transitions is often established during the early design stages, we have also collected low-fidelity versions of high-fidelity UI transitions. We aim to apply user simulation on both low-fidelity UI mockups and high-fidelity UI screens to improve UI design processes.

3. Data collection

We recruited 40 participants and conducted a two-part study to collect the data necessary for AI-driven user simulation of UI transitions. We prepared 1776 consecutive UI screen pairs for participants to annotate. The participants were asked to identify the link UI element between the consecutive screens and explain the reasons behind their choices. In the first part of the study, 140 pairs of UI screens were annotated. Each pair of UI screens received 20 annotations. The annotations from this process were used to create the test set. Additionally, the qualitative explanations left by the participants were analyzed to investigate how users commonly understand the connection between consecutive UIs and identify the UI elements for navigating between them. In the second part of the study, each UI screen was only annotated by one participant, allowing us to accumulate more annotated data for model training. A total of 1636 UI screens were annotated in this process and used as training data.

3.1. Sources and representations of UI screens

We sampled UI screens from the GUI animation sub-dataset of the Rico dataset (Deka et al., 2017) by extracting the start and end frames of each UI transition animation. We only sampled single-tap gestures as we focus on how users identify link UIs between two screens.

To explore effective user simulation methods for both high-fidelity and low-fidelity UI screens, we prepared three types of UI representations within our dataset: high-fidelity UI images, low-fidelity UI images, and JSON files containing UI metadata (id, positions, and text descriptions of UI elements, see Fig. A.9). The high-fidelity UI images were directly extracted from the Rico dataset, while the low-fidelity counterparts and JSON metadata were created manually by human creators (for the test set) or automatically by AI (for the training set). The rationale behind this approach was that manually created UI mockups or metadata though more accurate, were often in limited supply. Using high-quality data that was created manually for testing could ensure the evaluation accuracy, while training the model on less accurate UI data created by AI helped enhance the model's robustness and adaptability. Through this dataset processing, we aimed to reduce the quality requirements for the training data and explored the effectiveness of using automatically generated UI representations as training data.

When creating the test set, we recruited two professional UI designers to recreate the low-fidelity UI images according to their high-fidelity counterparts. Additionally, we hired a human annotator to manually annotate the positions and text descriptions of each UI element. Since the annotator was allowed to generate such metadata using generative AI, we also invited a human inspector to review the annotated metadata to ensure accuracy. The rigorous collection and annotations process of test data allowed the test set to become a benchmark dataset to measure the performance of user simulation models. When applying AI to automatically generate UI data in the training set, we removed color from the UI screenshots and applied YOLOv8[1] to detect UI elements on each screen at first. Then we extracted the position of each UI element and generated text descriptions for each UI element using GPT-4o.

3.2. User annotation

We recruited 40 participants (23 females, 17 males; aged 22 to 31, M = 26.98, SD = 2.08) to annotate their anticipated link UI elements between given UI screen pairs. All participants were college students with over five years of smartphone experience. As such, they were familiar with common UI semantics. They had no prior exposure to the apps sampled in the dataset, ensuring their reliance on general mobile UI conventions rather than app-specific knowledge to annotate the UI data.

Each participant was assigned a set of UI screen pairs consisting of a current screen and a target screen. Their task was to annotate the link UI element that they believe connects the two screens. If participants felt uncertain about the correct link UI element among other possible candidates, they could mark the UI pairs as "uncertain". After completing the annotation, they were also asked to provide a rationale for their choices. All annotations were performed using the same annotation tool, deployed on a laboratory workstation. 27 participants completed the annotation in person on this workstation, and 13 participants accessed it through remote-desktop software. The annotation sessions began with a brief standardized tutorial and written instructions. Participants could ask for help anytime during the annotation process except for task hints or correctness feedback.

The participants only annotated high-fidelity UI screens which reflect how users typically experience user interfaces in real smartphone usage. We translated the annotations to the corresponding low-fidelity UI images and JSON representations of the screens. This approach enables us to train models that can simulate users' possible choices of link UI elements when interacting with finalized UI prototypes even before high-fidelity prototypes are completed.

We adopted an unbalanced distribution strategy to maximize annotated data while collecting sufficient samples for analyzing users' common selection patterns and strategies. As such, the 140 test pairs received annotations from 20 individual participants during the first part of the collection study, while the 1,636 training data received a single annotation during the second part of the collection study.

3.3. Quantitative analysis of dataset

To enhance the test set's diversity and accuracy, we first analyzed the distribution of different navigation patterns and then refined the annotations of link UI elements and uncertainty through agreement analysis. To conduct the agreement analysis, we categorized the annotated link UI elements into five groups based on vote count: the most preferred, second preference, third preference, fourth preference, and others. Using Fleiss' kappa equation (Eq. (1)), we calculated an agreement score P_i for each pair, where n_{ij} represents the number of raters assigning the ith pair to the jth category, and n denotes the total votes per pair (n = 20).

$$P_i = \frac{1}{n(n-1)} \sum_{j=1}^{k} n_{ij}(n_{ij} - 1) \tag{1}$$

Table 1 shows the distribution of navigation patterns in the test set, which covers 7 common types (tab bar, float button, top navigation, bottom navigation, hamburger menu, vertical navigation, and card (Neil, 2014)). The agreement analysis among test set samples revealed that some samples failed to achieve high agreement scores, suggesting participants' uncertainty. Consequently, we labeled samples as "uncertain" if they were either directly annotated as such by more than half of the participants or had an agreement score below 0.3. Finally, 21 samples were annotated as "uncertain" in the test dataset. Among the remaining 119 samples, the average agreement score was 0.79 (SD = 0.23), reflecting substantial agreement on the link UI elements. The UI element receiving the most votes was designated as the users' anticipated link UI element.

As for the training set, agreement analysis was not conducted due to limited labels. Therefore, participants' annotations for uncertainty and link UI elements were directly used for model training. There were 750 "uncertain" samples and 936 "certain" samples in the training set, and the entire training set was used to train the model for "uncertainty prediction". Since participants only annotated their anticipated link UI elements on the data samples where they were sure about their choices, the subset where the data samples were annotated as "certain" was employed to train the link UI prediction model.

3.4. Qualitative findings

We conducted a qualitative analysis of participants' explanations for their selections during the annotation of test data, aiming to uncover common strategies they used to identify the link UI element. After filtering out vague or non-informative responses (*e.g.*, "I chose this part"), we retained 2,766 effective codes. Through thematic analysis, we derived 5 common themes (denoted as W1 to W5) that characterize how users commonly understand navigation mechanisms and recognize the link UI element for screen transitions. Furthermore, we explored how these common ways of identifying link UI elements contribute to user uncertainty during UI transitions.

W1: Leveraging Semantic consistency. Experienced UI designers usually obey the semantic consistency principles for users to easily relate consecutive UI screens. In practice, both the link UI element and the following screen should contain similar semantics (e.g., a button named "Plans" leading to a screen titled "Plans" in Fig. 1(a)).

W2: Following UI content hierarchy. UI content in a mobile app is often hierarchical. The home screen often displays icons of various functions, while children screens, deeper in the hierarchy, present the details of specific functions. Users can often decide to go back or move forward based on their understanding of the hierarchical relationship between consecutive UI screens. For example, in the transition illustrated in Fig. 1(e), participants commonly indicated they should return to the main page.

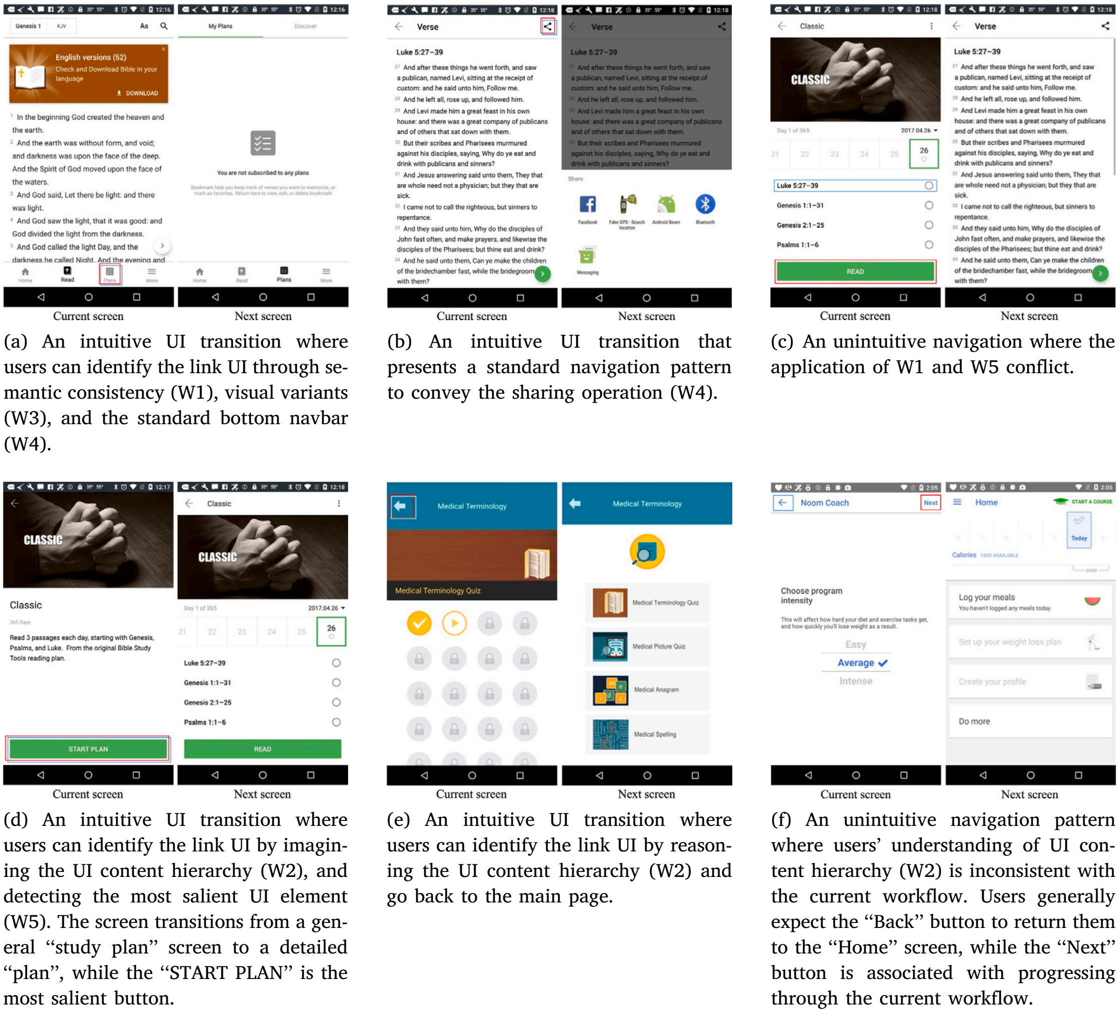
W3: Detecting visual variants between two UI screens. Visual feedback is presented to confirm that interaction happened with the UI element (e.g., in Fig. 1(a), the "Plans" button changes color between screens).

W4: Recognizing standard navigation patterns. Through years of smartphone use, users have learned common navigation patterns. Familiar components like icons, menus, or navigation bars placed in expected locations serve as intuitive cues (*e.g.*, Fig. 1(b) shows a sharing icon at the top-right corner of the screen, a familiar and intuitive feature. Fig. 1(a) also shows a standard bottom nav bar).

[1] https://universe.roboflow.com/encora-inc/ui-detection-with-yolov8

Table 1
Count distribution of navigation patterns in the test set.

Design pattern	Tab bar	Float button	Top navigation	Bottom navigation	Side menu	Vertical navigation	Card	Total
Count	18	29	16	20	14	28	15	140



(a) An intuitive UI transition where users can identify the link UI through semantic consistency (W1), visual variants (W3), and the standard bottom navbar (W4).

(b) An intuitive UI transition that presents a standard navigation pattern to convey the sharing operation (W4).

(c) An unintuitive navigation where the application of W1 and W5 conflict.

(d) An intuitive UI transition where users can identify the link UI by imagining the UI content hierarchy (W2), and detecting the most salient UI element (W5). The screen transitions from a general "study plan" screen to a detailed "plan", while the "START PLAN" is the most salient button.

(e) An intuitive UI transition where users can identify the link UI by reasoning the UI content hierarchy (W2) and go back to the main page.

(f) An unintuitive navigation pattern where users' understanding of UI content hierarchy (W2) is inconsistent with the current workflow. Users generally expect the "Back" button to return them to the "Home" screen, while the "Next" button is associated with progressing through the current workflow.

Fig. 1. Examples of UI transitions to exemplify users' common ways to understand UI transitions and identify the link UI element. The red bounding box shows the real link UI set by designers while the blue bounding box shows users' common choice of the link UI.

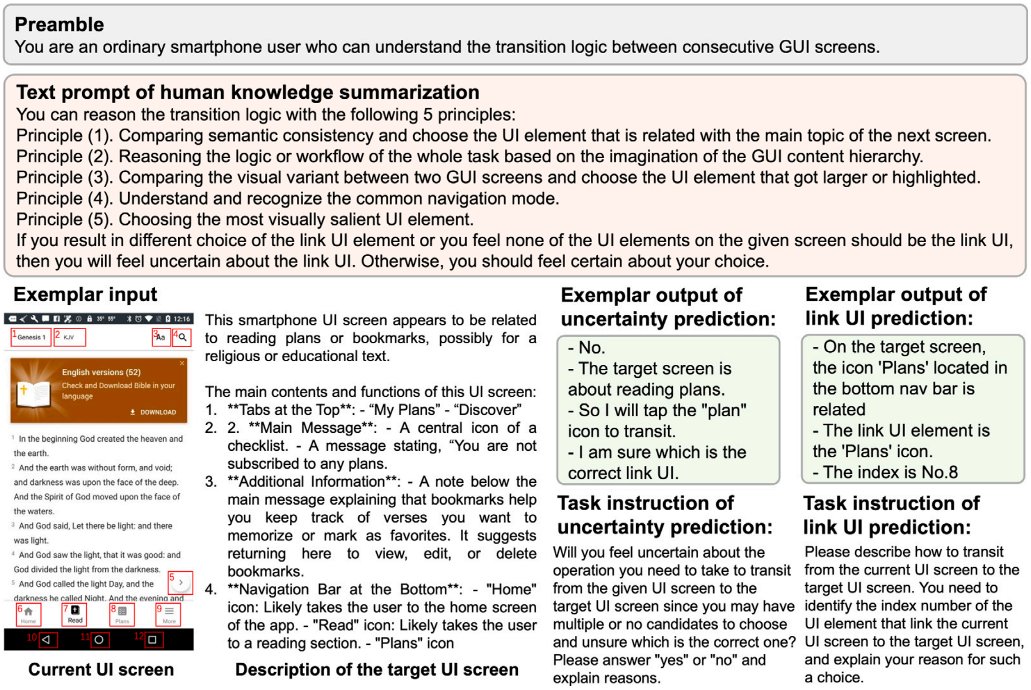
W5: Confirmation through visual saliency. When a UI screen contains only a few UI elements, the most visually salient element tends to capture the users' attention. This makes users more confident in their selection of the link UI. For example, when identifying the link UI between the screen pairs shown in Fig. 1(d), participants frequently noted that "START PLAN" is the only button, leading them to believe it was the correct choice.

Conflicts between common ways of understanding UI transitions can lead to user uncertainty. Although intuitive UI transition logic can indicate the appropriate link UI element to the user in different ways, ambiguous interfaces can lead to conflicts between these interpretations. In the example shown in Fig. 1(c), 12 participants selected the list item "Luke 5:27-39" based on consistent text label semantics (W1), while 6 participants chose the "Read" button due to its visual saliency (W5). Additionally, when a user workflow is inconsistent with the hierarchical relationship reflected in the GUI semantics (W2), users may become confused. As Fig. 1(f) shows, the designer set the "Next" button as the link UI element because "Next" aligns with the program's workflow semantics (W1). However, users expected the "Back" button to link to the "Home" page, following their perception of the UI hierarchy (W2).

4. AI-driven user simulation models

Based on the dataset and human factor insights we collected in Section 3, we develop AI-driven user simulation models to assist UI/UX designers in formative evaluations of UI transitions. Specifically, we focus on two tasks:

1. Uncertainty prediction: Predicting and explaining how uncertain users feel about selecting the UI element on the current screen to reach the target screen.
2. Link UI prediction: Predicting the UI element users would most likely select to transition between screens and explaining the reason behind the prediction.



Preamble

You are an ordinary smartphone user who can understand the transition logic between consecutive GUI screens.

Text prompt of human knowledge summarization

You can reason the transition logic with the following 5 principles:
Principle (1). Comparing semantic consistency and choose the UI element that is related with the main topic of the next screen.
Principle (2). Reasoning the logic or workflow of the whole task based on the imagination of the GUI content hierarchy.
Principle (3). Comparing the visual variant between two GUI screens and choose the UI element that got larger or highlighted.
Principle (4). Understand and recognize the common navigation mode.
Principle (5). Choosing the most visually salient UI element.
If you result in different choice of the link UI element or you feel none of the UI elements on the given screen should be the link UI, then you will feel uncertain about the link UI. Otherwise, you should feel certain about your choice.

Exemplar input

1 In the beginning God created the heaven and the earth.
2 And the earth was without form, and void; and darkness was upon the face of the deep. And the Spirit of God moved upon the face of the waters.
3 And God said, Let there be light: and there was light.
4 And God saw the light, that it was good: and God divided the light from the darkness.
5 And God called the light Day, and the darkness he called Night. And the evening and

Current UI screen

This smartphone UI screen appears to be related to reading plans or bookmarks, possibly for a religious or educational text.

The main contents and functions of this UI screen:
1. **Tabs at the Top**: - "My Plans" - "Discover"
2. 2. **Main Message**: - A central icon of a checklist. - A message stating, "You are not subscribed to any plans.
3. **Additional Information**: - A note below the main message explaining that bookmarks help you keep track of verses you want to memorize or mark as favorites. It suggests returning here to view, edit, or delete bookmarks.
4. **Navigation Bar at the Bottom**: - "Home" icon: Likely takes the user to the home screen of the app. - "Read" icon: Likely takes the user to a reading section. - "Plans" icon

Description of the target UI screen

Exemplar output of uncertainty prediction:

- No.
- The target screen is about reading plans.
- So I will tap the "plan" icon to transit.
- I am sure which is the correct link UI.

Task instruction of uncertainty prediction:

Will you feel uncertain about the operation you need to take to transit from the given UI screen to the target UI screen since you may have multiple or no candidates to choose and unsure which is the correct one? Please answer "yes" or "no" and explain reasons.

Exemplar output of link UI prediction:

- On the target screen, the icon 'Plans' located in the bottom nav bar is related
- The link UI element is the 'Plans' icon.
- The index is No.8

Task instruction of link UI prediction:

Please describe how to transit from the current UI screen to the target UI screen. You need to identify the index number of the UI element that link the current UI screen to the target UI screen, and explain your reason for such a choice.

Fig. 2. The illustration of exemplified prompting elements. Five types of prompting elements can appear in a prompt: preamble, text prompt of human knowledge summarization, exemplar input, exemplar output, and task instructions. One prompt can involve one or more types of prompting elements.

4.1. Problem formulation

In typical UI transition scenarios, users form an internal representation of their target screen. They interpret the current screen's UI semantics and search for the UI element that links it to the target screen. To computationally simulate this cognitive process, we structure the model input to include both the current UI screen and a text description of the target screen. With such formulation, we aim to predict whether users are likely to feel uncertain about selecting the correct UI element for transitioning to the target screen, as well as identifying their common choice of the link UI element.

We use a textual description rather than directly providing the visual image of the target screen. This choice serves two important purposes. On one hand, it prevents the model from completing the task through trivial visual feature matching alone, forcing it to deeply understand the semantic connotations of various elements (including labels, icons, and functional affordances) within the current screen. On the other hand, this formulation also mirrors common practice in design evaluation workflows, where evaluators are given goal descriptions rather than visual representations of the target screen. Text-based target descriptions not only provide a unified and controlled way to express user intent but also avoid prematurely exposing the visual information of the target screen, thereby preventing disruptive biases from influencing the process of searching for relevant UI elements.

4.2. Prompting GPT-4o as baseline

We first consider GPT-4o, which has demonstrated state-of-the-art (SOTA) performance on various multimodal tasks, to serve as the baseline for our target tasks: (1) predicting users' uncertainty regarding UI transitions and generating explanations for these predictions, and (2) predicting the link UI element needed for transitioning to the target UI screen, along with the corresponding explanations. We apply the following prompt elements, as summarized in Fig. 2.

Preamble The preamble introduces the context, capabilities, and role of the LLM, guiding it to perform the given task as the intended role. We provide the LLM with the following preamble:

"You are an ordinary smartphone user who can understand the transition logic between consecutive GUI screens".

Summarization of human knowledge To guide the LLM in identifying the link UI element based on common user logic, we pass a summary of relevant human knowledge derived from our findings in the data collection study Section 3.

"You can reason the transition logic with the following 5 principles:

Principle (1). Comparing semantic consistency and choose the UI element that is related with the main topic of the next screen.

Principle (2). Reasoning the logic or workflow of the whole task based on the imagination of the GUI content hierarchy.

Principle (3). Comparing the visual variant between two GUI screens and choose the UI element that got larger or highlighted.

Principle (4). Understanding and recognizing the common navigation mode.

Principle (5). Choosing the most visually salient UI element.

If you result in a different choice of the link UI element or you feel none of the UI elements on the given screen should be the link UI, then you will feel uncertain about the link UI. Otherwise, you should feel certain about your choice".

Few-shot exemplars To construct the prompt, we use four exemplars based on the dataset we collected during the data collection study Section 3. Each exemplar consists of an input, which includes a UI screen along with the text description of the target screen, and an expected output specifying the index number of the link UI element and the anticipated explanation of the reasoning process, as shown in Fig. 2. The exemplars provided during prompting adhere to the same schema as the input screens used for prediction. For instance, when GPT-4o is tasked with predicting link UI elements based on low-fidelity UI images, the exemplars embedded in the prompt are likewise presented in low-fidelity form.

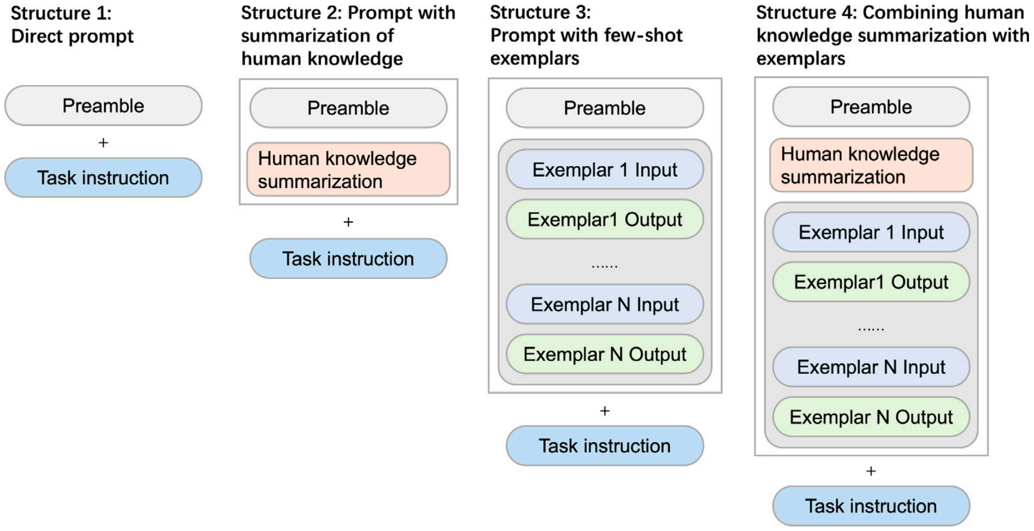

Structure 1: Direct prompt
Preamble
+
Task instruction
Structure 2: Prompt with summarization of human knowledge
Preamble
Human knowledge summarization
+
Task instruction
Structure 3: Prompt with few-shot exemplars
Preamble
Exemplar 1 Input
Exemplar1 Output
......
Exemplar N Input
Exemplar N Output
+
Task instruction
Structure 4: Combining human knowledge summarization with exemplars
Preamble
Human knowledge summarization
Exemplar 1 Input
Exemplar1 Output
......
Exemplar N Input
Exemplar N Output
+
Task instruction

Fig. 3. There are four types of prompt structure. (1) The direct prompt, which includes only a preamble and a task instruction. (2) The prompt that incorporates human knowledge summarization, where the human knowledge is added to the prompt. (3) The prompt that utilizes few-shot exemplars, where a set of exemplar inputs and outputs is included. (4) The prompt structure that combines human knowledge summarization with exemplars, involving both the human knowledge summary and exemplars in the prompt.

Task instruction We provide the multimodal LLM with the current UI screen and the text description of the target UI screen, along with the following instructions to introduce the intended task:

- Uncertainty prediction task: *"Will you feel uncertain about the operation you need to take to transit from the given UI screen to the target UI screen since you may have multiple or no candidates to choose from and are unsure which is the correct one? Please answer 'yes' or 'no' and explain the reasons".*
- Link UI prediction task: *"Please describe how to transit from the current UI screen to the target UI screen. You need to identify the index number of the UI element that links the current UI screen to the target UI screen and explain your reason for such a choice".*

We develop four prompting strategies for GPT-4o to perform uncertainty prediction and link UI prediction (see Fig. 3), and generate explanations for the prediction results as follows:

- Direct prompt: Only augment the task instruction with a preamble.
- Prompt with summarization of human knowledge: Augmenting the task instruction by combining preamble with human knowledge summarization.
- Prompt with few-shot exemplars: Augmenting the task instruction by combining the preamble with a few pairs of input and output exemplars.
- Combining human knowledge summarization with exemplars: Augmenting the task instruction by combining preamble, human knowledge summarization, and a few pairs of input and output exemplars together.

4.3. *Uncertainty prediction: Binary classification*

In addition to prompting LLMs, we frame uncertainty prediction as a binary classification task. By leveraging a pre-trained CLIP model, which aligns visual and linguistic features in a unified space, we train a binary classifier to predict users' uncertainty when navigating to the target UI screen.

4.3.1. *Uncertainty prediction pipeline*

Fig. 4 provides an overview of the uncertainty prediction model pipeline. The current UI screen and the text description of the target screen are first encoded by a pre-trained CLIP model with a ViT-L/14 backbone (Radford et al., 2021). The embeddings for each screen are then concatenated and passed through three fully connected layers. A Sigmoid activation function in the output layer produces a binary classification, indicating whether users might feel uncertain about the UI transition between the given pair of consecutive screens.

4.3.2. *Training procedure*

We trained the uncertainty prediction model with the training dataset collected in Section 3. First, we only used high-fidelity UI screens to train the model, while evaluating it on both the high-fidelity and low-fidelity test data. We then trained an uncertainty prediction model on both the high-fidelity data and low-fidelity versions of our collected training dataset. The models were implemented using PyTorch and trained by minimizing binary cross-entropy loss. The training was conducted on an NVIDIA A100, with a learning rate of 1e-3, using a batch size of 256 screen pairs, and optimized with the Adam optimizer (Kingma and Ba, 2014). The models trained over different epochs were selected using cross-validation on both high-fidelity and low-fidelity data.

4.3.3. *Explanation generation*

To generate explanations for the uncertainty prediction results, we prompt GPT-4o by providing the prediction results within the task instruction. The task instruction is: *"You are certain/uncertain about the UI element you need to tap to transition from the given screen to the target screen. Please explain your reasoning".* Additionally, we experiment with the four prompting strategies discussed in Section 4.2.

4.4. *Link UI prediction: Fine-tuning LLaMa-3.1-8B*

Given the limited transparency of commercial LLMs, we aim to enable open-source alternatives for link UI prediction. To minimize the computing resource requirements, we focus on memory-efficient models. In this work, we fine-tune the LLaMa-3.1-8B model, which

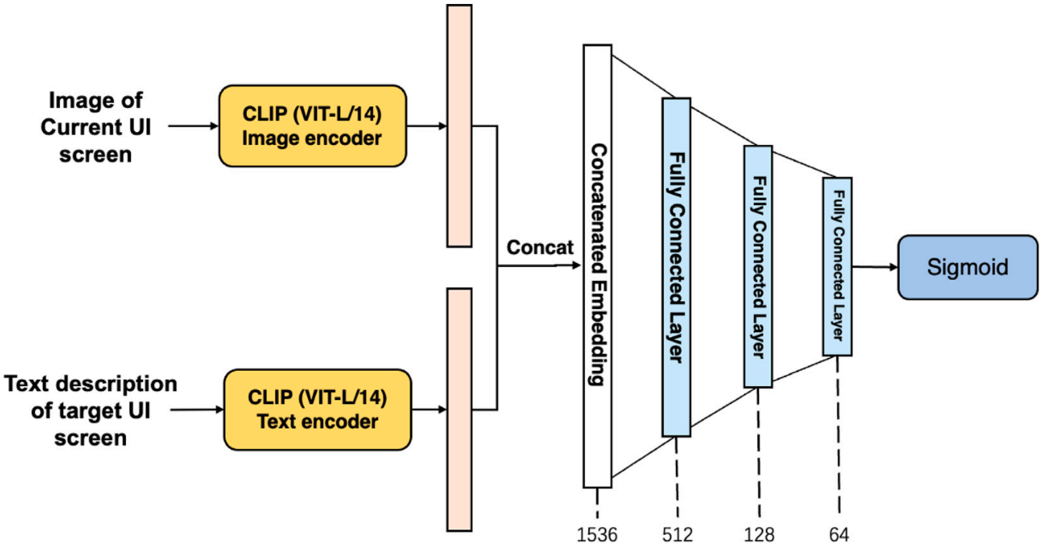


Image of Current UI screen
CLIP (VIT-L/14) Image encoder
Text description of target UI screen
CLIP (VIT-L/14) Text encoder
Concat
Concatenated Embedding
Fully Connected Layer
Fully Connected Layer
Fully Connected Layer
Sigmoid
1536
512
128
64

Fig. 4. The pipeline of the uncertainty prediction model.

is an open-source, memory-efficient LLM developed by Meta.[2] Its low VRAM requirements allow it to run off consumer-grade GPUs (starting at the low-cost NVIDIA RTX 4070) and have demonstrated SOTA performance among open-source models on various tasks. Based on the JSON representation of UI screens, we format the training dataset in a JSON structure compatible with LLaMa-3.1:

```
{
    "Instruction": task instruction,
    "Input": {
        "current screen": JSON representation
             of the current UI screen,
        "target screen": text description of
            the target UI screen
    },
    "Output": {
        "id": id number,
        "reason": explanation for the choice
            of the link UI
    }
}
```

We use the LLaMa factory (Zheng et al., 2024) to fine-tune LLaMa-3.1 on our training set. For this process, we apply 4-bit QLoRA (Dettmers et al., 2024) to the base model, integrating LoRA modules into all linear layers with a rank of 8. The learning rate is set to 5e-5, with a warm-up ratio of 0.1, and the model is fine-tuned on an A100 GPU with a batch size of 2. During inference, we set the temperature to 0.001 to ensure consistent generation. Additionally, we evaluate four different prompting strategies, as we did with GPT-4o (see Section 4.2), on both the fine-tuned and original LLaMa-3.1 models.

4.5. Technical evaluation of uncertainty prediction and link UI prediction

We evaluate the performance of uncertainty prediction models and link UI prediction models using the benchmark test set described in Section 3, consisting of 140 data samples.

4.5.1. Performance on uncertainty prediction

UI pairs where users were uncertain about the transition logic were labeled as positive, while pairs where users were confident in how to navigate from one screen to the next were labeled as negative, leading to 21 positive and 119 negative samples.

We compare our uncertainty prediction models against GPT-4o. As shown in Table 2, two uncertainty prediction models trained with different data adoption strategies both outperform GPT-4o in predicting user uncertainty. Specifically, the uncertainty prediction model trained exclusively on high-fidelity UI screens achieves 35.00% precision and 66.67% recall for positive samples when tested on low-fidelity data. The model that was trained on both high-fidelity and low-fidelity UI screens achieves 26.92% precision and 66.67% recall. In contrast, GPT-4o's best performance on low-fidelity data reaches only 22.58% precision and 47.62% recall. When tested on high-fidelity data, the model trained solely on high-fidelity UI screens achieves a precision of 27.08% and a recall of 61.90% for positive samples. The model trained on both types of UI screens achieves a precision of 22.58% and a recall of 66.67%. By contrast, GPT-4o's best performance on high-fidelity data is 20.83% precision and 23.81% recall.

Our experiments reveal that GPT-4o continues to struggle with identifying UI transitions that could lead to navigation difficulties for users. Given that designers prioritize detecting unintuitive transitions over detecting intuitive ones, it is critical to achieve high recall for positive samples—those that cause user uncertainty—while maintaining high precision for negative samples. By ensuring high recall for positive samples and high precision for negative samples, our models are more effective than GPT-4o in pre-screening UI transitions for designers.

Moreover, models trained solely on high-fidelity UI data serve as effective zero-shot learners for low-fidelity UI data. These models are able to successfully transfer their understanding of high-fidelity interfaces to low-fidelity screens, achieving even better performance on low-fidelity data than on high-fidelity data. Through comparing models trained on different datasets, we found that incorporating machine-generated low-fidelity UI data into the training set does not significantly enhance the model's capabilities; in fact, it can even degrade performance. Therefore, when manually created low-fidelity UI data is limited, training models directly on high-fidelity UI data and applying them to low-fidelity UI prediction tasks is a more effective strategy than generating more low-fidelity data by automatically reducing the fidelity of high-fidelity screens.

Finally, we observe a consistent limitation across all models: they exhibit low precision in identifying positive samples. This means that

[2] https://ai.meta.com/blog/meta-llama-3-1/

Table 2
Comparison of model performance on uncertainty prediction with the test set.

	Low fidelity				High fidelity			
	Negative samples		Positive samples		Negative samples		Positive samples	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Our Model 1	**93.00%**	78.15%	**35.00%**	**66.67%**	**91.30%**	70.59%	**27.08%**	61.90%
Our Model 2	92.05%	68.07%	26.92%	**66.67%**	91.03%	59.66%	22.58%	**66.67%**
Prompt 1	84.89%	**99.16%**	0.00%	0.00%	85.00%	**100.00%**	0.00%	0.00%
Prompt 2	81.73%	73.95%	20.51%	47.62%	86.20%	84.03%	20.83%	23.81%
Prompt 3	87.16%	79.83%	22.58%	33.33%	85.59%	84.87%	18.18%	19.04%
Prompt 4	85.19%	57.98%	15.25%	42.85%	87.13%	73.95%	20.51%	38.06%

Notes:
Our Model 1: Uncertainty prediction model (trained on PURE high-fidelity UI screens).
Our Model 2: Uncertainty prediction model (trained on BOTH high-fidelity and low-fidelity UI screens).
Prompt 1: Directly prompt GPT-4o.
Prompt 2: Prompt GPT-4o with human knowledge summarization.
Prompt 3: Prompt GPT-4o with few-shot exemplars.
Prompt 4: Combine human knowledge with few-shot exemplars to prompt GPT-4o.

UI transitions predicted as "uncertain" may not always indicate actual user uncertainty. As a result, designers need to examine these cases further, which can increase their workload. The GPT-4o's weakness in this aspect could be rooted in reinforcement learning (RL) strategies, which are common in LLMs' training. Rewarding decisive, helpful answers while penalizing hedging (*e.g.*, "I'm not sure") could make the model less likely to accurately simulate user uncertainty. Our work presents collecting UI transition samples that specifically challenge users' confidence, and training a dedicated binary classifier for uncertainty prediction can be a practical remedy. Such an uncertainty predictor can be exposed to LLM-driven agent systems through function calling. Although our uncertainty predictor outperforms GPT-4o in correctly identifying positive samples, its precision on the benchmark test set is still insufficient. We see two main causes: (1) the overall size of our training data is limited, and (2) when annotating positive–negative pairs, each data sample only reflects the uncertainty judgment from a single annotator, introducing label noise and bias. Collecting a larger volume of high-quality, carefully annotated training data, ideally with multiple raters per example and stronger coverage of positive samples, is essential for further improving the performance of the uncertainty prediction model.

4.5.2. *Performance on link UI prediction*

We conducted experiments with different prompting methods on GPT-4o, the original LLaMa-3.1-8B, and our fine-tuned LLaMa-3.1-8B using the benchmark test set. Additionally, we replicated UXAgent (Lu et al., 2025) using GPT-4o as a foundation model, which is the most relevant open-source user simulation framework for UI interactions. We excluded samples labeled "uncertain" from training the link UI prediction model because they lack annotations for the link UI element. For prompting GPT-4o, we used three input formats: the low-fidelity version, the high-fidelity version, and the JSON representation. In the case of LLaMa and UXAgent, we only used the JSON representation of UI screens.

Accuracy was measured by comparing the model's predictions with the users' common choices. To further evaluate performance, we classified each sample as positive if the model's prediction matched the ground truth link UI element defined by the designers, and negative if it did not. Predictions that aligned with human labels were considered true predictions, while those that did not were categorized as false predictions. As such, the accuracy, precision, and recall of the models are shown in Table 3.

As shown in Table 3, using few-shot exemplars as prompts for GPT-4o was more effective than summarizing human knowledge. However, for LLaMa, text summarization of human knowledge performed better than few-shot examples. Additionally, prompting GPT-4o with a JSON representation of UI screens, which describes the position and semantics of UI elements in text form, resulted in better performance compared to using low-fidelity or high-fidelity UI screenshots. This indicates that a textual representation of the positions and semantics of UI elements is an effective way to help large language models (LLMs) predict the link UI element. This approach could serve as a viable alternative to low-fidelity UI mockups.

Fine-tuning LLaMa-3.1-8B significantly improved its performance on the link UI prediction task. Combined with an appropriate prompting method, the fine-tuned LLaMa-3.1-8B achieved comparable results to GPT-4o, with greater transparency, lower operational requirements, and the guarantee that the model will operate as expected over time.

Finally, while there is room for improvement in both accuracy and recall, all models demonstrated relatively high precision. This indicates that when the model's predicted link UI element aligns with the designers' expected element, there is a strong likelihood that users will also consider it the correct link UI. This quality highlights the potential of prompting LLMs to effectively inspect the intuitiveness of UI transitions.

4.6. *Human evaluation of explanation generation*

To evaluate the credibility, clarity, and designers' satisfaction of the generated explanations for the predicted results, we conducted a human evaluation for the generated explanations.

4.6.1. *Methods*

We invited 10 UI/UX design practitioners (5 females and 5 males; aged from 23 to 28, M = 25.70, SD = 1.77) to act as human evaluators tasked with assessing AI-generated explanations by comparing them to real user feedback on specific UI transitions. We selected 4 pairs of UI transitions where users were uncertain about the link UI element and 4 pairs where users typically identified the correct link UI element, using samples from the benchmark test set. Each sample corresponds to explanations from 20 users. Both user-provided explanations and AI-generated explanations were presented to the evaluators. They rated the explanations on a 5-point Likert scale based on three criteria: credibility (1 = least credible, 5 = most credible), clarity (1 = least clear, 5 = most clear), and satisfaction (1 = least satisfied, 5 = most satisfied).

Each evaluator first assessed the explanations provided by real users, followed by evaluating 8 versions of AI-generated explanations. For the UI pairs labeled as "uncertain", these 8 versions were created by combining 4 different prompting strategies (as shown in Fig. 3) with two conditions: (1) prompting GPT-4o using known uncertainty prediction results, and (2) prompting GPT-4o to manage both the

Table 3
Comparison of link UI prediction performance across different UI presentation, prompting methods, and foundation models.

	GPT-4o								
	Low fidelity			High fidelity			Json presentation		
	ACC	Precision	Recall	ACC	Precision	Recall	ACC	Precision	Recall
Method 1	46.96%	96.08%	45.79%	62.61%	97.14%	62.39%	66.09%	100.00%	63.89%
Method 2	47.83%	98.08%	46.37%	66.09%	97.34%	66.34%	42.61%	100.00%	40.54%
Method 3	50.43%	98.18%	49.09%	**70.43%**	97.44%	70.37%	**71.30%**	97.47%	71.30%
Method 4	**53.91%**	98.28%	52.29%	66.96%	98.63%	66.06%	69.57%	97.40%	69.44%
UXAgent	–	–	–	–	–	–	43.48%	98.04%	45.45%

	LLaMa-3.1-8B			Fine-tuned LLaMa-3.1-8B		
	Json presentation			Json presentation		
	ACC	Precision	Recall	ACC	Precision	Recall
Method 1	26.03%	96.43%	24.32%	55.65%	100.00%	55.56%
Method 2	33.04%	100.00%	31.25%	64.35%	98.59%	63.64%
Method 3	30.43%	96.67%	26.36%	56.52%	96.83%	55.96%
Method 4	33.04%	97.14%	30.91%	57.39%	96.88%	56.88%

Notes:
Method 1: Direct prompt.
Method 2: Prompt with human knowledge summarization.
Method 3: Prompt with few-shot exemplars.
Method 4: Combine human knowledge with few-shot exemplars.

Table 4
Subjective scores of generated explanations for uncertainty prediction.

	Real users	GPT-4o uncertainty prediction and explanation generation				GPT-4o pure explanation generation with already predicted uncertainty				Kruskal-Wallis (H, P)
		Method 1	Method 2	Method 3	Method 4	Method 1	Method 2	Method 3	Method 4	
Credibility	4.00 (1.00)	4.00 (0.00)	4.00 (0.00)	4.00 (1.25)	4.00 (2.00)	4.00 (1.00)	4.00 (1.00)	4.00 (1.00)	4.00 (1.00)	(9.71, 0.29)
Clarity	4.00 (1.00)	4.00 (1.25)	4.00 (1.00)	4.00 (2.00)	4.00 (1.00)	4.00 (1.25)	4.00 (2.00)	4.00 (2.00)	4.00 (2.00)	(5.94, 0.65)
Satisfaction	3.50 (1.25)	4.00 (1.25)	4.00 (1.00)	4.00 (1.00)	4.00 (1.00)	4.00 (1.00)	4.00 (1.00)	4.00 (2.00)	4.00 (1.00)	(7.28, 0.50)

Notes:
n = 40 for all evaluations.
Method 1: Direct prompt.
Method 2: Prompt with summarization of human knowledge.
Method 3: Prompt with few-shot exemplars.
Method 4: Combine human knowledge with exemplars.

uncertainty prediction and explanation generation. For the UI pairs where participants commonly identified the correct link UI element, the 8 versions of explanations were generated from a combination of the 4 prompting strategies (as detailed in Section 4.2 and shown in Fig. 3) along with two different models: GPT-4o and the fine-tuned LLaMa-3.1-8B.

4.6.2. Results

To ensure that our evaluation provides reliable and representative results, we first analyzed how the average standard deviation of reviewer ratings varies with the panel size (Fig. 5). For the three criteria: clarity, credibility, and satisfaction, the average standard deviation of scores gradually converges as the number of evaluators increases from two to ten. This indicates that a panel of ten evaluators provides representative estimates and is sufficient to mitigate bias induced by sample size.

The distribution of subjective scores for the different explanation generation methods suggested non-normality, which was confirmed by a Shapiro–Wilk test. As a result, we applied non-parametric tests to evaluate the significant differences between the explanation generation methods. As shown in Table 4, there was no significant difference between the various methods for generating explanations related to uncertainty prediction results. All AI-generated explanations received similar subjective scores in terms of credibility, clarity, and satisfaction with explanations from real users.

For explanations regarding the selection of the link UI element, Kruskal–Wallis tests revealed significant differences among the methods in terms of credibility ($p = 0.005 < 0.01$, see Table 5). Further analysis using Dunn's t-test with *p*-value adjustment showed that real users' explanations were rated significantly higher in credibility compared to those generated by GPT-4o with few-shot examplars, and those generated by fine-tuned LLaMa-3.1-8B, whether it was prompted directly, with human knowledge summarization, or examplars. However, Dunn's t-test also indicated that the clarity of real users' explanations and the satisfaction they received were not significantly different from that of AI-generated explanations.

The statistical results demonstrate that combining the binary classifier of uncertainty prediction with GPT-4o's explanation capability provides higher prediction accuracy while ensuring high explanation quality. For generating explanations related to link UI prediction results, GPT-4o performs on par with our fine-tuned LLaMa-3.1-8B.

5. Design of the AI-driven user simulation tool for UI transition inspection

We integrate the user simulation functions developed in Section 4 into an interactive tool named UTP (UI Transition Predictor). UTP facilitates formative UI transition inspection by simulating user decisions with AI to quickly identify unintuitive UI transitions.

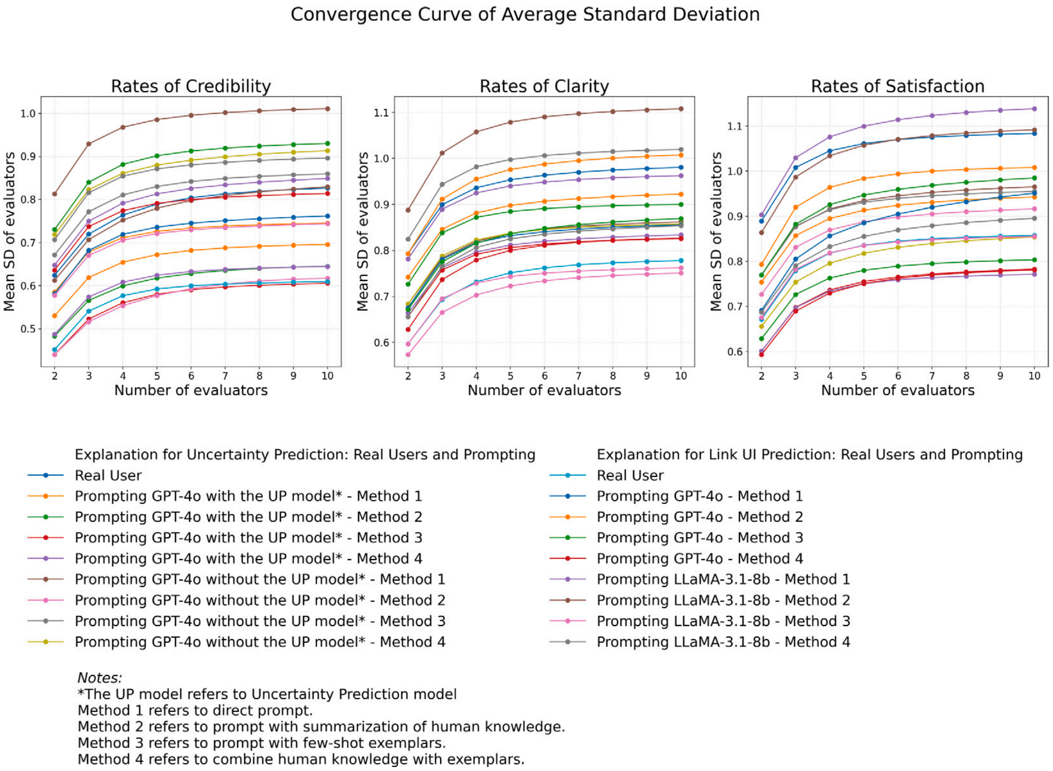
5.1. System pipeline and operation

UTP allows designers to input a UI flow composed of several screens and provides insights into the intuitiveness of UI transitions. Fig. 6 summarizes its pipeline. UTP first separates the UI flows into a series of

Table 5
Subjective scores of generated explanations for link UI prediction.

	Real users	GPT-4o				Fine-tuned LLaMa-3.1-8B				Kruskal–Wallis (H, P)
		Method 1	Method 2	Method 3	Method 4	Method 1	Method 2	Method 3	Method 4	
Credibility	5.00 (1.00)	4.00 (1.00)	4.00 (1.00)	4.00 (1.25)	4.00 (1.00)	4.00 (1.25)	4.00 (2.00)	4.00 (1.25)	4.00 (2.00)	(22.18, 0.005**)
Dunn's t test **P** (p-adjustment)		1.00	0.978	0.015*	1.00	0.013*	0.030*	0.038*	0.419	
Clarity	4.00 (2.00)	4.00 (1.00)	4.00 (1.25)	4.00 (2.00)	4.00 (1.00)	4.00 (2.00)	4.00 (1.00)	4.00 (2.00)	4.00 (1.00)	(12.12, 0.15)
Satisfaction	4.00 (1.00)	4.00 (1.00)	4.00 (2.00)	4.00 (2.00)	4.00 (1.00)	4.00 (2.00)	3.50 (1.00)	4.00 (2.00)	4.00 (2.00)	(12.88, 0.12)

Notes:
n = 40 for all evaluations.
Method 1: Direct prompt.
Method 2: Prompt with summarization of human knowledge.
Method 3: Prompt with few-shot exemplars.
Method 4: Combine human knowledge with exemplars.



Notes:
*The UP model refers to Uncertainty Prediction model
Method 1 refers to direct prompt.
Method 2 refers to prompt with summarization of human knowledge.
Method 3 refers to prompt with few-shot exemplars.
Method 4 refers to combine human knowledge with exemplars.

Fig. 5. The curves presenting the convergence of the average standard deviation.

screen pairs, each representing a UI transition step. Then, each screen pair is processed as follows: (1) the target screen is analyzed by the screen description module, which generates a text description of the UI screen, (2) the current screen is parsed by the UI metadata generator, which creates a JSON representation of the screen. Following this pre-processing step, UTP operates in two modes:

1. **Screening mode**: for automated screening of UI transitions.
2. **Dedicated mode**: designers manually annotate candidate UI elements that could be confusing to users, UTP then disambiguates and decides which one the user is likely to pick.

In the screening mode, the system leverages the uncertainty prediction model to predict whether the users may feel uncertain about the link UI element for each step. When the prediction result is "uncertain", the explanation generation module generates the explanation for such uncertainty and proposes a list of candidate UI elements that could be interpreted as a link UI element to the target screen, providing designers with detailed information on the source of ambiguity. When the prediction result is "certain", the system leverages the link UI prediction model to predict the link UI element and generate the explanations.

In the dedicated mode, we allow the designers to manually mark the UI elements they want to analyze and compare. Designers mark the coordinates of each element by drawing their bounding boxes onto the UI screen. The UI element selector removes the UI elements outside of the bounding boxes in the UI screen metadata represented with JSON. Then, the system transfers the edited UI screen metadata and the description of the target UI screen to the link UI prediction module to predict the link UI element and generate corresponding explanations.

5.2. Modules

To build the operations detailed above, UTP relies on following modules.

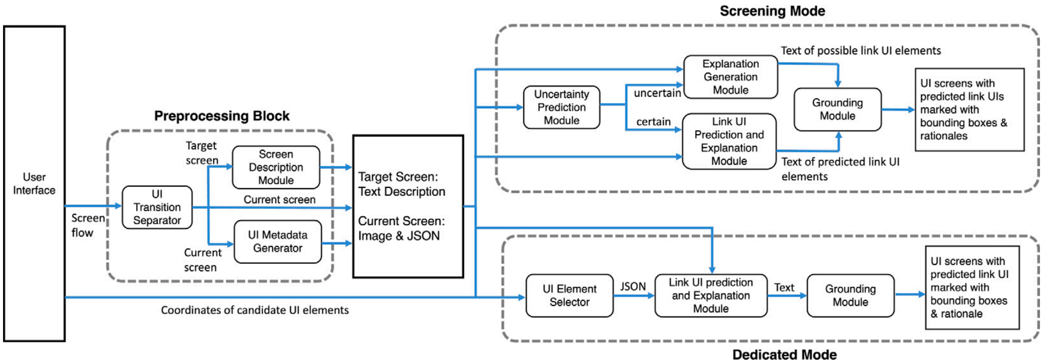

Screening Mode
Preprocessing Block
User Interface
Screen flow
UI Transition Separator
Target screen
Screen Description Module
Current screen
UI Metadata Generator
Current screen
Target Screen: Text Description
Current Screen: Image & JSON
Uncertainty Prediction Module
uncertain
certain
Explanation Generation Module
Link UI Prediction and Explanation Module
Text of possible link UI elements
Grounding Module
Text of predicted link UI elements
UI screens with predicted link UIs marked with bounding boxes & rationales
Coordinates of candidate UI elements
UI Element Selector
JSON
Link UI prediction and Explanation Module
Text
Grounding Module
UI screens with predicted link UI marked with bounding boxes & rationale
Dedicated Mode

Fig. 6. The system pipeline of the UI Transition Predictor.

5.2.1. *Pre-processing*

UI Transition Separator. The UI Transition Separator separates a sequence of UI flow into several pairs of consecutive UI screens, which are the basic units to the user simulation functions.

Screen Description Module. The screen description module generates descriptions of the primary content and functions of a given UI screen. We implement this by passing the UI image to GPT-4o, along with the task instruction *"Please generate the summary for the given smartphone UI. Describe the main contents or functions in this UI screen"*.

UI Metadata Generator. This module translates the UI screen into a JSON file which contains the position and semantic information of UI elements (as shown in Fig. A.9). We use the pre-trained YOLOv8[3] to detect the UI elements in a UI screen. Then we pass the image of detected UI elements to GPT-4o to generate the alt-text for the UI elements using the task instruction *"Please generate the alt text or caption for the given smartphone UI element. If it include text information, you should also include its text information in your caption. If it is a pure icon or image, you should generate the alt text or caption that describes its semantic meaning"*.

5.2.2. *Prediction*

Uncertainty Prediction Module. We use the uncertainty prediction model described in Section 4 to predict whether users may feel uncertain about the UI transition for the given step.

Explanation Generation Module. As described in Section 4, we provide GPT-4o with the current UI screen image, the description of the target UI screen, and the uncertainty prediction result to generate explanations for the uncertainty prediction. By using exemplars where users make different selections of the link UI element, GPT-4o is prompted to suggest potential candidate link UI elements.

Link UI prediction module. Based on the technical evaluation results, prompting GPT-4o with few-shot exemplars using the JSON representation of UI screens yielded the best performance for predicting the link UI element among the tested methods. As a result, we implemented this approach in the tool.

Grounding Module. Using the generated text content regarding the prediction of the link UI element, we prompt GPT-4o to map the predicted link UI element to its corresponding JSON object within the UI screen metadata. This allows us to retrieve the bounding box coordinates from the JSON file and visually mark the corresponding UI element on the user interface.

5.3. *User interface*

Fig. 7 details the user interface of UTP. Designers can drag a sequence of screenshots from their desktop to UTP as a UI flow. They can either directly click the "Predict" button to use the screening mode, or mark candidates they want to test in the dedicated mode before clicking the "Predict" button. The prediction result includes a sequence of UI screens, each with one or two bounding boxes to mark the possible link UI, and a paragraph of explanation for the prediction result. UTP also provides an assistance function for designers to remove the bounding box marks created by mistake (using the "Redo" button).

6. Evaluation user study: Evaluating the effectiveness of AI-driven user simulation in UTP for inspecting UI transitions

We conducted a user study to evaluate the effectiveness of UTP in helping designers inspect UI transitions. The evaluation aims to address the following research questions:

1. Does user simulation with UTP help designers accurately detect unintuitive UI transitions and confirm intuitive ones?
2. Does user simulation reduce the time required to inspect UI transitions?
3. What value and limitations do designers perceive in applying user simulation to UI transition inspection?

We ran a within-subject study in which each participant completed six UI transition inspection tasks under three tool conditions (without UTP, with the screening mode, with the dedicated mode). We evaluated the participants' inspection accuracy, time costs, and self-confidence under each condition. We also dig the usefulness and limitations of the AI-driven user simulation methods through short post interviews.

6.1. *Participants*

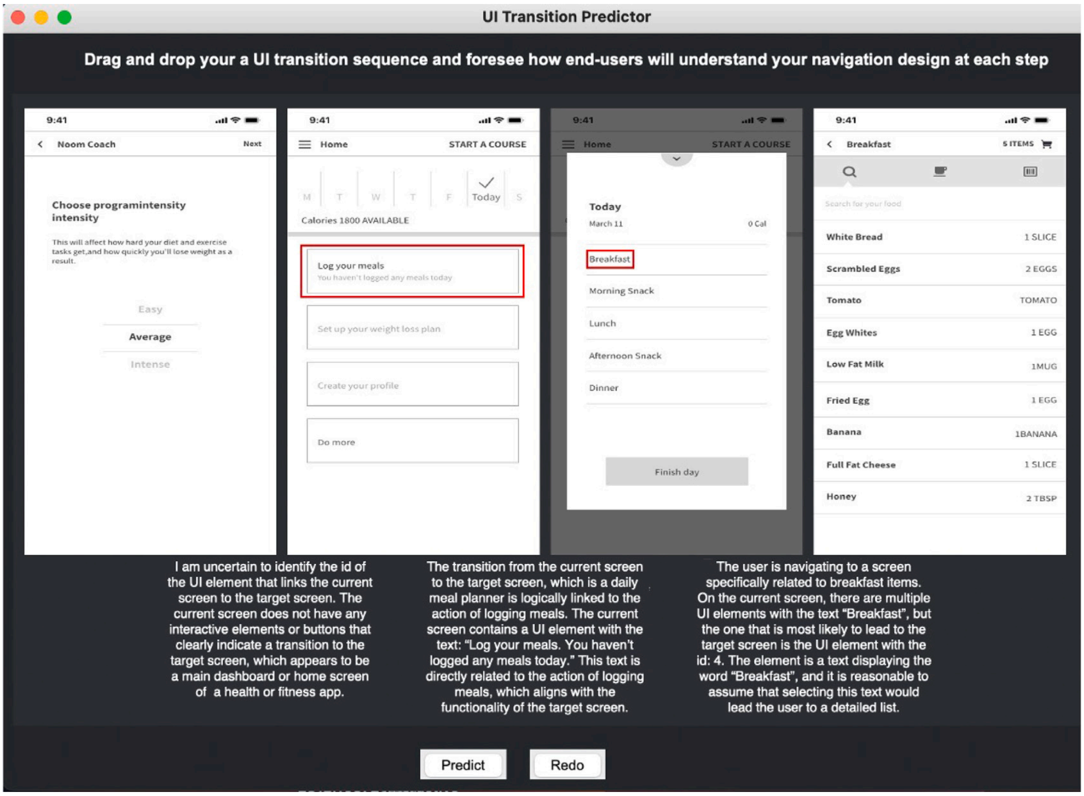
We recruited 12 participants (4 females and 8 males) through online social platforms and collected their demographic information. The participants were aged from 23 to 28 (M = 23.50, SD = 2.12). Six of them were industry UX design managers, three were UI designers or interns, and three were non-professionals who design mobile UIs for independent projects. They all held a college degree or higher. Demographic information of the participants is listed in Table B.7, Appendix. We provided the participants with 55 HKD for their participation.

6.2. *Materials and apparatus*

The six UI transition inspection tasks were based on high-fidelity UI flows extracted from the benchmark dataset (as outlined in Section 3). Each UI flow involved two transition steps. Three of these UI flows were logically intuitive, where users can select the link UI elements correctly. The other three contained design flaws, where users were uncertain or failed to select the correct link UI element.

Since the performance of the user simulation models was already quantified in Section 4, this user study examined whether accurate simulation can effectively assist designers during UI transition inspection. We ensure that any observed human performance or confidence

[3] https://universe.roboflow.com/encora-inc/ui-detection-with-yolov8



Drag and drop your a UI transition sequence and foresee how end-users will understand your navigation design at each step

I am uncertain to identify the id of the UI element that links the current screen to the target screen. The current screen does not have any interactive elements or buttons that clearly indicate a transition to the target screen, which appears to be a main dashboard or home screen of a health or fitness app.

The transition from the current screen to the target screen, which is a daily meal planner is logically linked to the action of logging meals. The current screen contains a UI element with the text: "Log your meals. You haven't logged any meals today." This text is directly related to the action of logging meals, which aligns with the functionality of the target screen.

The user is navigating to a screen specifically related to breakfast items. On the current screen, there are multiple UI elements with the text "Breakfast", but the one that is most likely to lead to the target screen is the UI element with the id: 4. The element is a text displaying the word "Breakfast", and it is reasonable to assume that selecting this text would lead the user to a detailed list.

Fig. 7. The user interface of UI Transition Predictor. Users can drag a UI flow to the interface. Directly clicking the "Predict" button can invoke screening mode. To use dedicated mode, users need to mark candidate UI elements first on each UI screen, then click "Predict" button. The predicted link UI elements will be marked with red bounding boxes. The explanations of AI's choices are presented in the middle of each UI screen pair. Users can click the "Redo" button to clear the AI-generated contents in the interface.

stems from the presentation and use of accurate simulation signals, rather than being confounded by model prediction errors. Therefore, we selected only UI flows for which the models either correctly predicted user uncertainty or accurately identified the user-anticipated link UI element as deliberately including cases with known model errors would add little value. This sampling choice did not imply that UTP could always return correct feedback. For UI flows predicted as user-uncertain, the screening mode would further propose possible candidates of users' choices, which may contain errors and be misleading. When using dedicated mode, users must first specify a set of candidate link UI elements and if the correct option were omitted, UTP would fail to produce a correct judgment. Therefore, UTP should be treated as decision support rather than an oracle and designers were expected to interpret its outputs and made their own choices, mirroring realistic usage.

The UTP prototype was deployed on a laboratory workstation. Three participants came to the lab and used the system on site to complete the experimental tasks, while nine completed the tasks remotely via remote desktop software. The prototype system automatically recorded the time each operation step took in the inspection process.

6.3. Procedure

Participants first received an overview of the study procedure and watched a short tutorial video instructing the UTP's screening mode and dedicated mode. Then, they practiced on tutorial examples until they were familiar with the system features before beginning the formal tasks. Each participant was required to inspect the assigned UI flows under three conditions.

- **C1: Without UTP.** Participants load the UI flow into the interface of the UTP. Without using the user simulation features provided by the UTP, they manually inspected each transition step, marked the link UI element users might choose, described the users' likely reasoning, and concluded whether the flow was intuitive.
- **C2: Screening mode.** After loading the UI flow, participants clicked "Predict" button to activate the screening mode, which simulated users' understanding of each UI transition step and predicted users' anticipated link UI element. Participants could accept or reject the AI prediction, optionally marking their own choice and rationale, and then concluded whether the flow was intuitive.
- **C3: Dedicated mode.** Participants first conducted a manual pass as in C1, and marked multiple candidates of link UI elements for each transition step. After that, they clicked "Predict" button and the dedicated mode would be invoked to examine the candidate UI elements they wanted to assess further. As in C2, they could accept or reject the model's output and documented a final judgment of the UI transition intuitiveness.

Each condition was assigned two UI transition flows (one is an intuitive UI flow and the other is an unintuitive UI flow), and the sequence of these three conditions was counterbalanced across participants using the Latin square method. The order of intuitive and unintuitive UI flows was randomized for each condition. We did not disclose to participants that the UI flow set was balanced, nor that each condition contained one intuitive and one unintuitive UI flow. The participants were required to judge each UI flow independently.

The design inspection documents from each participant were saved for further analysis. After completing each condition, participants were asked to rate their confidence in the inspection results on a 5-point Likert scale for each UI flow they inspected. They were also invited to a short post-interview about the usefulness and limitations of the tool. 9 of the participants participated in the post-interview.

Table 6
Accuracy of UI flow inspection.

	Without UTP	Screening mode	Dedicated mode
Overall accuracy	58.33% (14/24)	75.00% (18/24)	62.5% (15/24)
UI flows including design faults	41.67% (5/12)	75.00% (9/12)	33.33% (4/12)
Intuitive UI flows	75.00% (9/12)	75.00% (9/12)	91.67% (11/12)

To minimize variations due to factors like participants' writing habits, we only recorded and calculated the time taken to identify the link UI element. This time measurement began when participants started loading the UI flows and ended when they finished marking their predicted link UI element. The inference time of the AI models was excluded from the time cost calculations, as it depends on computing power and does not reflect the actual impact of the AI model's output on the design inspection process.

6.4. Quantitative results

We present the data collected and the metrics used for quantitative analysis. The inspection accuracy, time cost, and designers' confidence in design inspection are quantitatively analyzed.

6.4.1. Data collection

Through the experiment, we got 72 completion time records (12 participants × 2 UI flows of each condition × 3 conditions), with 24 time records per condition. We coded the participants' design inspection documents and rated their inspection accuracy. For the three UI flows with unintuitive transitions, we analyzed whether participants successfully identified the design defects and highlighted the unintuitive steps within the UI flow. In total, 36 data points (12 participants × 3 unintuitive UI flows) were collected. For the three intuitive UI flows, we evaluated the accuracy of participants' predictions for users' choice of the link UI element. Since each UI flow consisted of multiple transition steps, a prediction was considered correct only if participants accurately predicted the link UI element at each step and identified the flow as intuitive. Again, 36 data points (12 participants × 3 intuitive UI flows) were collected to assess prediction accuracy for users' UI transition choices. Additionally, we gathered 72 records of subjective feedback regarding participants' confidence in the design inspection of UI flows (1 = least confident, 5 = most confident).

6.4.2. Accuracy

Overall result. As summarized in Table 6, screening mode improves overall accuracy relative to the condition without UTP (75.00% vs. 58.33%). Confirmation mode yields intermediate overall accuracy (62.50%).

Post-hoc analysis by UI kinds. As shown in Table 6, under the condition without UTP and the dedicated mode, participants often fail to identify the unintuitive navigation design. The accuracy is only 41.67% without UTP, and 33.33% with the dedicated mode. With the screening mode of UTP, participants can achieve an accuracy of 75.00% in identifying unintuitive UI flows. For intuitive UI flows, participants sometimes felt uncertain and suggested alternative link UI choices, resulting in an accuracy of 75.00%. Meanwhile, the dedicated mode helped participants confidently make the correct choice, improving accuracy to 91.67%.

Adoption of AI-simulated user choices. We coded participants' acceptance of the AI-predicted user choices. Participants ignored AI predictions in **5/24** screening trials and **6/24** dedicated inspections.

Implications. Our results indicate that the screening mode is suitable for flaw discovery (75.00% on unintuitive flows vs. 41.67% without UTP and 33.33% in the dedicated mode), while the dedicated mode is useful at verifying flows that appear correct (91.67% on intuitive flows vs. 75.00% screening mode and without UTP). This pattern suggests a staged workflow: designers could first screen candidate UI flows to surface potential defects quickly, and use dedicated inspection to confirm those that would have already been regarded as intuitive. Because dedicated inspection underperforms on unintuitive UI flows, it should not be used as a stand-alone detector.

We also found that overrides were infrequent but costly. In the screening mode, participants overrode the AI in 5/24 trials and every override resulted in an error; in the dedicated inspection, 6/24 overrides occurred, 4 of which were erroneous. Qualitative annotations suggest a plausible mechanism: even when the uncertainty prediction was correct, the explanation sometimes surfaced low-probability alternatives to illustrate ambiguity, which appears to have induced skepticism and triggered overrides. When designers and the AI diverged, tentative rationales lacked persuasive force, particularly when the link UI prediction is correct. Rather than encouraging blind acceptance of predictions from AI, the tool should promote calibrated reliance where explanations should adapt to model confidence so they persuade when warranted and hedge when not.

6.4.3. Time cost

The average time cost is 99.22s (SD = 55.88) without UTP, 83.96s (SD = 59.56) with dedicated mode, and 34.06s (SD = 29.72) with screening mode. The time cost distribution suggests non-normality, confirmed by a Shapiro–Wilk test. As shown in Fig. 8(a), Friedman's test with Bonferroni post hoc correction reveals that the time cost of screening mode is significantly less than the condition without UTP (time reduced by 65.70%, $p < 0.001$), and with dedicated mode (time reduced by 59.43%, $p < 0.001$). The results validate that AI-driven user simulation can reduce the time cost for UI transition inspection.

6.4.4. Confidence

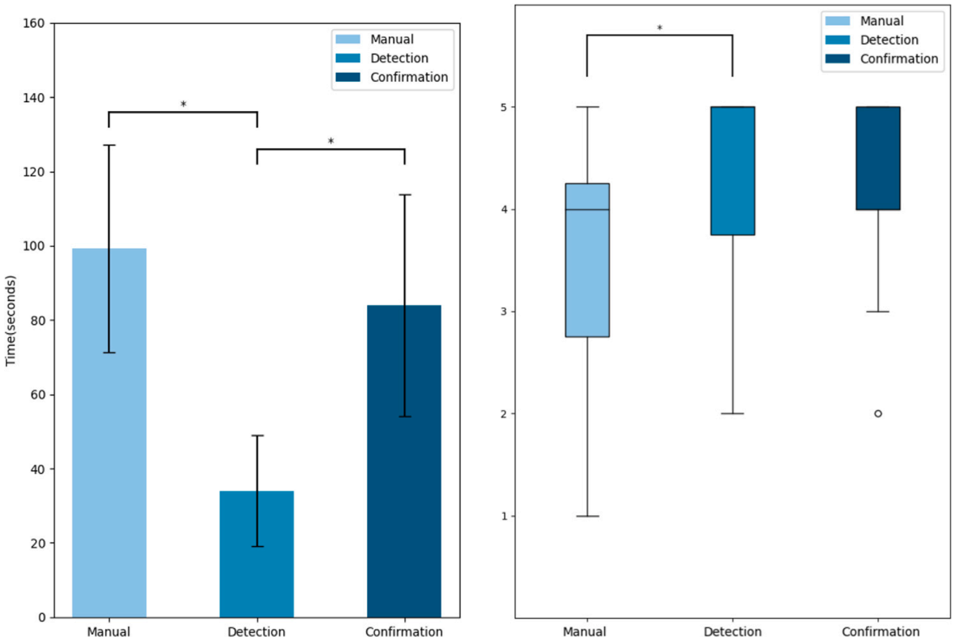
Fig. 8(b) shows a box plot presenting subjective scores received by the condition without UTP, screening mode, and dedicated mode in terms of enhancing the confidence of UI/UX practitioners. The distribution of the ratings is non-normal, confirmed by a Shapiro–Wilk test. The Friedman test confirms the significant difference among these three conditions (without UTP: Median = 4, IQR = 1.5; screening mode: Median = 5, IQR = 1.25; dedicated mode: Median = 4, IQR = 1, $p < 0.05$). This confirms the significant effect that AI-driven user simulation brings to the designers in enhancing their confidence. Post hoc pairwise test with Bonferroni correction further confirms that the screening mode can significantly improve the designers' confidence in the design inspection ($p < 0.05$).

6.5. Qualitative results

Our qualitative analysis explores the perceived value of AI-driven user simulation and areas where participants suggested improvements.

6.5.1. Usefulness of AI-driven user simulation: reminding and inspiring

Participants generally found the AI models effective in reflecting user behavior and providing meaningful explanations. Although they strongly believed in their expertise so that they would not directly adopt the prediction results, they agreed that the UTP offers extra protection for UI transition design in addition to common design workflows. They noted that the AI enhanced design evaluation by reminding potential flaws and inspiring new perspectives. For instance, some participants mentioned that the AI made unexpected suggestions they had not considered (P3) and reminded them to rethink potentially misleading UI elements in their designs (P11).



(a) Bar chart of time cost on different modes.

(b) Box plot of subjective score in terms of confidence

Fig. 8. Statistical plot of time cost and confidence score.

6.5.2. *Limitations and implications*

While participants acknowledged the benefits of AI-driven user simulation, they also identified limitations and suggested areas for improvement.

Improving the trust and providing concrete suggestions. Participants generally agreed that while the user simulation results can be effective, they still had difficulties in fully trusting them. To this end, two participants requested more quantitative data and visualizations to help them assess the credibility of predictions. As P8 stated, *"I want to know the percentage of users who would choose each of the candidates, that would be more helpful and convincing"*. Additionally, designers often use user simulation results as a reference for improving UI design. Therefore, they believe that actionable suggestions derived from the user simulation results could be more beneficial than the user simulation results themselves. As P11 indicated *"Currently it only analyzes the problem of the given design, I hope it also propose suggestions or recommendations to improve the design"*.

Focusing on assisting the design of ToB applications. ToB applications typically refer to applications that facilitate business transactions, focusing on the interactions and processes between businesses. These applications usually manage standard operation flows such as invoicing, order processing, and supply chain management. In contrast, ToC applications usually refer to applications that support commercial transactions between businesses and consumers. These applications are more customer-focused and content-oriented, such as e-commerce, online shopping, and video watching. Our findings suggest that the design of our UTP should be oriented toward supporting ToB applications rather than ToC applications. Participants indicated that ToB products, which are task-oriented, are more suitable for our tool. P7 suggested *"I think it might be more suitable for B-end tool-type products. For C-end products that need to guide user traffic, it doesn't seem quite appropriate, because users don't have a specific intention beforehand; they are led there"*. P1 indicated the gap between the current implementation of UTP and the real situations in designing ToB applications, *"B-end data samples usually containing prior knowledge of certain industries are often scarce. Since public datasets may not contain this kind of knowledge, further validation is needed"*.

7. Discussion

This section presents ethical considerations of applying AI-driven user simulation for UI evaluation, insights gained from our user studies, and novel methodological findings for simulating user decisions when navigating among mobile UIs. We also identify the limitations of the current research and outline directions for future work.

7.1. *Considerations of applying AI-driven user simulation for UI evaluation*

Currently, applying AI-driven user simulation is still an open research question in the field of human–computer interaction, presenting both opportunities and challenges. Our work provides empirical insights into the effectiveness and limitations of such simulations through the development and evaluation of UTP. Based on our research, we discuss the validity of using AI as simulated users, and the role of AI-driven user simulation plays in the UI evaluation.

7.1.1. *Validity of using AI as simulated users*

Based on our research findings from Section 4, directly prompting general LLMs like GPT-4o can introduce biases compared to real user feedback. However, by incorporating common thought patterns of users into AI models, we can mitigate these biases to some extent. We develop uncertainty prediction models that achieve higher precision (35.00% vs. 26.92% on low-fidelity data and 27.08% vs. 22.58% on high-fidelity data) and recall (66.67% vs. 47.62% on low-fidelity data and 66.67% vs. 38.06% on high-fidelity data) than GPT-4o in identifying UI transitions that may cause user uncertainty. This indicates that specialized models, trained on human-annotated datasets, can more accurately reflect user uncertainty compared to general-purpose LLMs. Additionally, with appropriate prompting strategies, GPT-4o's accuracy on link UI prediction can be improved from 42.61% to 71.30%, while through supervised fine-tuning on a specific human-annotated dataset, the accuracy of open-source LLaMa-3.1-8B can be improved from 33.04% to 64.35%. Therefore, by embedding the cognitive processes users employ when selecting link UI elements into LLMs through prompting or instruction-tuned datasets, we can effectively enhance the models' ability to mimic user decision-making patterns. This approach also aids in the miniaturization of user simulation models and facilitates localized deployment. Finally, as what we have conducted in this work, it is crucial to have a benchmark dataset annotated by multiple users to evaluate the performance of AI-simulated users. Once validated, AI models can serve as useful tools for design evaluation to some extent. We will release the benchmark dataset that we collected in this research.

7.1.2. The role of AI-driven user simulation in the UI evaluation

In the design evaluation and iteration process, the primary decision-makers are the designers themselves. The qualitative findings we derived from the evaluation user study reveals that the role of AI-driven user simulation should be to remind potential flaws or provide insights, helping designers avoid design errors caused by oversight or lack of consideration. AI-generated user feedback cannot replace real user experiments. Designers should still base their decisions on actual user testing, especially when the AI does not agree with designers' dedicated choices. In practical design scenarios, conducting user experiments can often be inconvenient (Xiang et al., 2024). Even low-cost informal user tests can consume significant time and effort from designers. Moreover, when the number of participants in user experiments is insufficient, the results can also be biased. Therefore, the greatest advantage of AI-driven user simulation lies in its convenience. It allows designers to obtain valuable feedback from a user perspective anytime and anywhere.

7.2. Summary of insights from user studies

Our data collection study established the foundation for simulating user behaviors on UI transitions. The benchmark dataset, annotated by 20 real users, captures common navigation patterns, ensuring both rigor and representativeness. We identified five general approaches users take in understanding UI transition logic: (1) Leveraging Semantic consistency. (2) Following UI content hierarchy. (3) Detecting visual variants between two UI screens. (4) Recognizing standard navigation patterns. (5) Confirmation through visual saliency. These insights are instrumental in guiding LLMs to simulate user behaviors effectively.

Our evaluation study shows the effectiveness of the screening mode of UTP, which improves the designers' accuracy of inspecting UI transitions from 58.33% (without UTP) to 75% while reducing the time cost by 65.70%. The dedicated mode should not be used as a stand-alone inspection method due to its low accuracy for detecting unintuitive UI transitions (33.33%). However, it can be support designers to confidently validate their decision and avoid misjudging the intuitive design of UI transitions due to its high accuracy on intuitive UI flows (91.67%). As a result, UTP can be adopted through a staged design workflow: screening to surface potential issues first, then confirming UI flows that are likely intuitive.

Interestingly, the credibility of AI-generated feedback is crucial in the design evaluation process. As shown in the user evaluation study, even when the prediction of user behavior was correct, the AI-generated explanations sometimes illustrate ambiguity, which may lead the design users to override correct predictions. An ideal yet challenging scenario involves generating more persuasive feedback when the AI accurately predicts user behavior, and less persuasive feedback when it does not, necessitating the AI model's ability to assess its own confidence in its predictions. Furthermore, to better meet designers' needs, the user simulation tool should offer quantitative analyses and visualizations that help assess the credibility of predictions, as well as provide concrete design recommendations.

7.3. Methods for simulating user decision in the UI understanding domain

We discuss the insights of two aspects in simulating user decision-making: predicting uncertainty and forecasting actions with link UI elements, which represent different decision states. Link UI prediction anticipates user actions, while uncertainty prediction assesses user confidence during navigation.

LLMs excel at predicting and explaining specific user actions but often fall short in capturing users' subjective uncertainty. This is likely because they tend to provide confident outputs due to extensive training. To tackle this issue, we model the uncertainty using traditional classification models with built-in self-confidence mechanisms that better reflect user hesitation. Based on the predicted uncertainty, LLMs can generate explanations accordingly. This hybrid approach opens up possibilities for exploring how LLMs and traditional models can work together to predict subjective confidence in various contexts. Additionally, a model trained on high-fidelity UI data can also effectively predict outcomes for low-fidelity UI mockups. This suggests an opportunity to utilize the automatically collected user interaction trace data on commercial apps to predict user uncertainty on designated mobile UI transitions, even when the model input consists of low-fidelity UI mockups during the inference. A key gap remaining to be addressed is the automatic detection of user uncertainty when using smartphone apps.

For link UI prediction, using a JSON representation of the UI screen with GPT-4o is more effective than using the raw UI image. Since GPT models primarily process text, structured formats like JSON allow for more effective reasoning by bridging the gap between vision and language. The JSON is created by YOLOv8, which detects UI elements, and GPT-4o labels them. This "agentic workflow" combines vision and language models, enhancing visual reasoning tasks in UI design. Currently, vision language models (VLMs) are evolving rapidly, and such an agentic workflow could be replaced by one-shot prediction. However, the JSON representation of UI screens, which can convey the semantic information of both high-fidelity and low-fidelity UIs, may still be effective in supporting design inspections at both early and later stages. Moreover, although the JSON data for training the user simulation model was automatically generated by AI models based on the UI screen images and does not match the quality of manually processed benchmark datasets, it has been effective for fine-tuning LLMs to predict user-anticipated link UI elements. This finding also suggests the possibility of predicting users' possible selection of link UI elements during a designated mobile UI transition by training the prediction model with the user interaction trace data automatically collected on commercial apps.

7.4. Limitations and future work

We discuss the limitations of this work and outline potential directions to address them, as well as opportunities to extend the current research further. The discussion of the limitations and future work is presented in three aspects: data collection, user study, and model development.

7.4.1. Data collection

For annotating low-fidelity UIs, we directly transferred the annotations from high-fidelity UIs rather than requiring participants to select link UI elements on low-fidelity screens. Our motivation is to enable the predictive model to simulate user selections on high-fidelity UIs based on a given low-fidelity UI flow. However, the validity of this approach requires further investigation, as UI elements in low-fidelity and high-fidelity UIs are not equivalent due to variations in affordance, saliency, and color. For example, if the design of a low-fidelity UI contains structural or semantic flaws that are partially compensated by the color information in its high-fidelity counterpart, the annotations from the high-fidelity data may conceal these deficiencies in the low-fidelity design. The presence of such cases in the dataset could potentially make the predictive model less sensitive to design flaws in low-fidelity UIs. In future work, we plan to explore these issues by collecting and comparing user annotations on both low-fidelity and high-fidelity UIs. We also plan to evaluate the performance of models trained on these two different annotation sets.

Considering that the performance of our uncertainty prediction model is hindered by the limited size of our training data and the reliance on annotations from a single annotator per sample, which introduces noise and bias. In the future, we plan to gather a larger volume of high-quality UI transition samples with multi-rater annotations to better capture the subtleties of user uncertainty.

Additionally, the data used in this study is sourced from the RICO dataset, which, while referring to older applications' UI screens, proved useful for our research. Since RICO was created when mobile UI design was still developing, it offered the opportunity to identify design flaws that aided our research. Moving forward, we plan to collect more recent datasets that reflect modern UI navigation paradigms, ideally drawn from iterative design drafts rather than finalized apps. We will also seek to gather more data from different specific user groups or persons, such as older adults or people with lower digital literacy, and examine whether such user simulations facilitate the accessibility of UI design.

7.4.2. User study

In the user study for collecting annotated UI data, participants were located in different places. Some participants joined in person, while others participated remotely via remote desktop. Although the annotation tool and the process remained consistent across locations, the mixed participant settings may introduce potential risks to data validity. In the evaluation study, each participant was assigned one intuitive UI flow and one unintuitive UI flow in each of the three conditions. Although the order of intuitive and unintuitive UI flows was randomized for each condition, there remains a possibility that participants might guess the intuitiveness of subsequent UI flows based on their evaluations of the previous ones.

7.4.3. Model development

The model development includes several simplifications. We focused exclusively on tap gestures, omitting other navigation gestures, and limited our prediction model to consecutive UI screens, ignoring the potential influence of previous user flows on users' decisions. Future work will address these limitations by incorporating more complex real-world smartphone interactions to enhance the model's generalizability.

Moreover, due to constraints of computing resources, we relied on a smaller open-source LLM for link UI prediction without integrating multimodal methodology. Combining visual cues from images with semantic and positional data from JSON could enhance prediction robustness, but bring more computing costs. Future work will investigate fusion-based approaches to leverage the strengths of both modalities. Despite these limitations, our results are promising, and we anticipate that fine-tuning larger open-source LLMs and multimodal LLMs will further improve performance in future research.

8. Conclusion

This work highlights the significant potential of applying AI-simulated user decision-making and behaviors to support the formative evaluation of UI transitions. Our contributions include a novel dataset and effective methods that advance AI-driven user simulation. Notably, our uncertainty prediction model outperforms GPT-4o, while a fine-tuned LLaMa-3.1-8B model achieves comparable results in predicting user-anticipated link UI elements. Technical evaluations reveal that current large language models still exhibit shortcomings in accurately simulating user behavior. Incorporating real user behavior data and decision-making strategies, either through model training or advanced prompting, can enhance the user simulation performance. Based on our user simulation method, we design and evaluate the UTP, an interactive tool which support designers to rapidly screen design flaws in UI flows and confirm the design that already appears to be intuitive. User evaluations validate that simulated user feedback guides and reminds designers in detecting potential design issues, but also acknowledge the limitations of current simulation techniques, which do not fully capture the credibility of real user interactions. Collecting more high-quality user data is critical for future improvements.

CRediT authorship contribution statement

Xiaozhu Hu: Writing – original draft, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Xiaoyu Mo:** Writing – original draft, Visualization, Formal analysis, Data curation. **Xiaofu Jin:** Writing – review & editing, Validation, Investigation. **Yuan Chai:** Software. **Yongquan Hu:** Validation, Resources. **Mingming Fan:** Writing – review & editing, Supervision, Resources. **Tristan Braud:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used GPT4 in order to proofread the paper. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded, in part, by project Z1010 from HKUST Center for Aging Science.

Appendix A. Example of the low-fidelity, high-fidelity and JSON representation version of data samples in the dataset

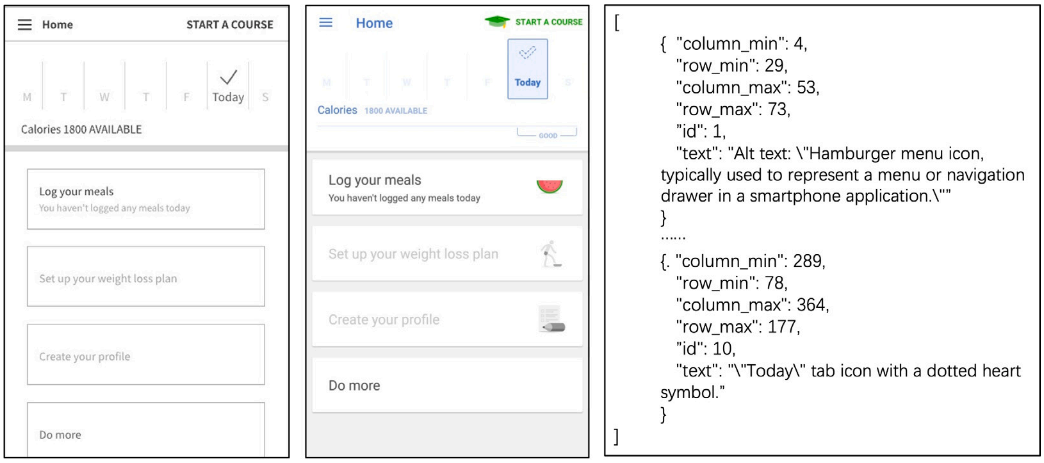
See Fig. A.9.

Appendix B. Demographic information of participants in the evaluation user study

See Table B.7.

Data availability

Data will be made available on request.



```
[
    { "column_min": 4,
      "row_min": 29,
      "column_max": 53,
      "row_max": 73,
      "id": 1,
      "text": "Alt text: \"Hamburger menu icon,
    typically used to represent a menu or navigation
    drawer in a smartphone application.\""
    }
    ......
    {. "column_min": 289,
      "row_min": 78,
      "column_max": 364,
      "row_max": 177,
      "id": 10,
      "text": "\"Today\" tab icon with a dotted heart
    symbol."
    }
]
```

Low-fidelity sample High-fidelity sample JSON representation

Fig. A.9. Example of the low-fidelity, high-fidelity and JSON representation of UI mockup.

Table B.7
Demographic information of participants.

Participant ID	Gender	Age	Literacy	Years of UX/UI design experience	Expertise
p1	Male	26	Bachelor's	6	Company UX Project Manager
p2	Female	25	Master's	2	Company UX Project Manager
p3	Female	23	Bachelor's	3	Interaction Design Student
p4	Male	27	PhD	4	Independent App Designer and Developer
p5	Male	27	PhD	4	Independent App Designer and Developer
p6	Male	26	PhD	3	Independent App Designer and Developer
p7	Male	25	Master's	2	Company UX Project Manager
p8	Male	27	Bachelor's	3	UI Designer
p9	Male	22	Bachelor's	3	UI Design Intern
p10	Female	28	PhD	6	Company UX Project Manager
p11	Male	28	Master's	6	Company UX Project Manager
p12	Female	27	Master's	6	Company UX Project Manager

References

Arning, K., Ziefle, M., 2009. Effects of age, cognitive, and personal factors on PDA menu navigation performance. Behav. Inf. Technol. 28 (3), 251–268.

Baeg, J., Hirahara, A., Fukazawa, Y., 1994. A development strategy of user navigation systems and gui applications. In: Proceedings Eighteenth Annual International Computer Software and Applications Conference. COMPSAC 94, IEEE, pp. 163–168.

Barr, P., Biddle, R., Noble, J., 2002. A taxonomy of user-interface metaphors. In: Proceedings of the SIGCHI-NZ Symposium on Computer-Human Interaction. pp. 25–30.

Bellamy, R., John, B., Kogan, S., 2011. Deploying CogTool: integrating quantitative usability assessment into real-world software development. In: Proceedings of the 33rd International Conference on Software Engineering. pp. 691–700.

Biswas, P., Robinson, P., 2010. Evaluating the design of inclusive interfaces by simulation. In: Proceedings of the 15th International Conference on Intelligent User Interfaces. pp. 277–280.

Blackmon, M.H., Mandalia, D.R., Polson, P.G., Kitajima, M., 2007. Automating usability evaluation: Cognitive walkthrough for the web puts LSA to work on real-world HCI design problems. Handb. Latent Semant. Anal. 345–375.

Burrell, A., Sodan, A.C., 2006. Web interface navigation design: which style of navigation-link menus do users prefer? In: 22nd International Conference on Data Engineering Workshops. ICDEW'06, IEEE, 42–42.

Bylinskii, Z., Kim, N.W., O'Donovan, P., Alsheikh, S., Madan, S., Pfister, H., Durand, F., Russell, B., Hertzmann, A., 2017. Learning visual importance for graphic designs and data visualizations. In: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology. pp. 57–69.

Candiasa, I.M., Gunadi, I.G.A., Putra, I.N.W.S., 2023. UX evaluation using firstclick, performance measurement, RTA, and questionnaire on E-commerce website. Sink.: J. Dan Penelit. Tek. Inform. 7 (1), 451–460.

Cepeda, C., Dias, M.C., Rindlisbacher, D., Gamboa, H., Cheetham, M., 2021. Knowledge extraction from pointer movements and its application to detect uncertainty. Heliyon 7 (1).

Chang, E., Dillon, T.S., 1998. The navigational aspects of the logical design of user interfaces. In: Proceedings First International Symposium on Object-Oriented Real-Time Distributed Computing. ISORC'98, IEEE, pp. 425–430.

Chen, C.F., Pistoia, M., Shi, C., Girolami, P., Ligman, J.W., Wang, Y., 2017. UI X-Ray: Interactive mobile UI testing based on computer vision. In: Proceedings of the 22nd International Conference on Intelligent User Interfaces. pp. 245–255.

Chen, J., Swearngin, A., Wu, J., Barik, T., Nichols, J., Zhang, X., 2022. Towards complete icon labeling in mobile applications. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. pp. 1–14.

Deka, B., Huang, Z., Franzen, C., Hibschman, J., Afergan, D., Li, Y., Nichols, J., Kumar, R., 2017. Rico: A mobile app dataset for building data-driven design applications. In: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology. pp. 845–854.

Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L., 2024. Qlora: Efficient finetuning of quantized llms. Adv. Neural Inf. Process. Syst. 36.

Djonov, E., 2007. Website hierarchy and the interaction between content organization, webpage and navigation design: A systemic functional hypermedia discourse analysis perspective. Inf. Des. J. 15 (2), 144–162.

Dørum, K., Garland, K., 2011. Efficient electronic navigation: A metaphorical question? Interact. Comput. 23 (2), 129–136.

Duan, P., Warner, J., Li, Y., Hartmann, B., 2024. Generating automatic feedback on UI mockups with large language models. In: Proceedings of the CHI Conference on Human Factors in Computing Systems. pp. 1–20.

Farkas, D.K., Farkas, J.B., 2000. Guidelines for designing web navigation. Tech. Commun. 47 (3), 341–358.

Feiz, S., Wu, J., Zhang, X., Swearngin, A., Barik, T., Nichols, J., 2022. Understanding screen relationships from screenshots of smartphone applications. In: Proceedings of the 27th International Conference on Intelligent User Interfaces. IUI '22, Association for Computing Machinery, New York, NY, USA, pp. 447–458. http://dx.doi.org/10.1145/3490099.3511109.

Fosco, C., Casser, V., Bedi, A.K., O'Donovan, P., Hertzmann, A., Bylinskii, Z., 2020. Predicting visual importance across graphic design types. In: Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology. pp. 249–260.

Granollers, T., Lorés, J., 2004. Incorporation of users in the evaluation of usability by cognitive walkthrough. In: HCI Related Papers of Interacción 2004. Springer, pp. 243–255.

He, Z., Sunkara, S., Zang, X., Xu, Y., Liu, L., Wichers, N., Schubiner, G., Lee, R., Chen, J., 2021. Actionbert: Leveraging user actions for semantic understanding of user interfaces. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 5931–5938.

Hong, W., Wang, W., Lv, Q., Xu, J., Yu, W., Ji, J., Wang, Y., Wang, Z., Dong, Y., Ding, M., et al., 2024. Cogagent: A visual language model for gui agents. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14281–14290.

Jiang, Y., Schoop, E., Swearngin, A., Nichols, J., 2023. Iluvui: Instruction-tuned LangUage-vision modeling of UIs from machine conversations. arXiv preprint arXiv:2310.04869.

Johns, C.A., Barz, M., Sonntag, D., 2023. Interactive link prediction as a downstream task for foundational gui understanding models. In: German Conference on Artificial Intelligence (Künstliche Intelligenz). Springer, pp. 75–89.

Kalbach, J., 2007. Designing Web Navigation: Optimizing the User Experience. "O'Reilly Media, Inc.".

Kim, H., Hornbæk, K., Lee, B., 2022. Quantifying proactive and reactive button input. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. pp. 1–18.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Koch, J., Oulasvirta, A., 2016. Computational layout perception using gestalt laws. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. pp. 1423–1429.

Kosinski, M., 2023. Theory of mind might have spontaneously emerged in large language models. arXiv preprint arXiv:2302.02083.

Lakoff, G., Johnson, M., 2008. Metaphors We Live By. University of Chicago Press.

Lee, C., Kim, S., Han, D., Yang, H., Park, Y.W., Kwon, B.C., Ko, S., 2020. GUIComp: A GUI design assistant with real-time, multi-faceted feedback. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–13.

Leesutthipornchai, P., Pradubsuwun, D., 2023. Quantitative evaluation of navigation controllers on mobile applications. In: 2023 11th International Conference on Information and Education Technology. ICIET, IEEE, pp. 61–65.

Lewis, C., Wharton, C., 1997. Cognitive walkthroughs. In: Handbook of Human-Computer Interaction. Elsevier, pp. 717–732.

Li, G., Li, Y., 2022. Spotlight: Mobile ui understanding using vision-language models with a focus. arXiv preprint arXiv:2209.14927.

Li, Y., Li, G., He, L., Zheng, J., Li, H., Guan, Z., 2020. Widget captioning: Generating natural language description for mobile user interface elements. arXiv preprint arXiv:2010.04295.

Li, Q., Luximon, Y., 2023. Navigating the mobile applications: The influence of interface metaphor and other factors on older adults' navigation behavior. Int. J. Human–Computer Interact. 39 (5), 1184–1200.

Lu, J., Srivastava, S., Chen, J., Shrestha, R., Acharya, M., Kafle, K., Kanan, C., 2024. Revisiting multi-modal LLM evaluation.

Lu, Y., Yao, B., Gu, H., Huang, J., Wang, Z.J., Li, Y., Gesi, J., He, Q., Li, T.J.J., Wang, D., 2025. UXAgent: An LLM agent-based usability testing framework for web design. http://dx.doi.org/10.1145/3706599.3719729.

Mahatody, T., Sagar, M., Kolski, C., 2010. State of the art on the cognitive walkthrough method, its variants and evolutions. Intl. J. Human–Computer Interact. 26 (8), 741–785.

Neil, T., 2014. Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps. "O'Reilly Media, Inc.".

Nielsen, J., 1992. Finding usability problems through heuristic evaluation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 373–380.

Nielsen, J., 1994. Usability inspection methods. In: Conference Companion on Human Factors in Computing Systems. pp. 413–414.

Nielsen, J., 1999. The top ten new mistakes of web design. p. 2001, Retrieved August 26.

Nielsen, J., Molich, R., 1990. Heuristic evaluation of user interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 249–256.

Polson, P.G., Lewis, C., Rieman, J., Wharton, C., 1992. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. Int. J. Man-Mach. Stud. 36 (5), 741–773.

Punchoojit, L., Hongwarittorrn, N., et al., 2017. Usability studies on mobile user interface design patterns: a systematic literature review. Adv. Human-Computer Interact. 2017.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., 2021. Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. PmLR, pp. 8748–8763.

Rawles, C., Li, A., Rodriguez, D., Riva, O., Lillicrap, T., 2024. Androidinthewild: A large-scale dataset for android device control. Adv. Neural Inf. Process. Syst. 36.

Schoop, E., Zhou, X., Li, G., Chen, Z., Hartmann, B., Li, Y., 2022. Predicting and explaining mobile ui tappability with vision modeling and saliency analysis. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. pp. 1–21.

Si, C., Zhang, Y., Yang, Z., Liu, R., Yang, D., 2024. Design2Code: How far are we from automating front-end engineering? arXiv:2403.03163.

Swearngin, A., Li, Y., 2019. Modeling mobile interface tappability using crowdsourcing and deep learning. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. pp. 1–11.

Tidwell, J., 2010. Designing Interfaces: Patterns for Effective Interaction Design. "O'Reilly Media, Inc.".

Vanderdonckt, J., Zen, M., Vatavu, R.D., 2019. AB4web: An on-line A/B tester for comparing user interface design alternatives. Proc. ACM Human-Computer Interact. 3 (EICS), 1–28.

Wang, B., Li, G., Li, Y., 2023. Enabling conversational interaction with mobile ui using large language models. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. pp. 1–17.

Wang, B., Li, G., Zhou, X., Chen, Z., Grossman, T., Li, Y., 2021. Screen2words: Automatic mobile UI summarization with multimodal learning. In: The 34th Annual ACM Symposium on User Interface Software and Technology. pp. 498–510.

Wang, J., Xu, H., Ye, J., Yan, M., Shen, W., Zhang, J., Huang, F., Sang, J., 2024. Mobile-agent: Autonomous multi-modal mobile device agent with visual perception. arXiv preprint arXiv:2401.16158.

Webster, J., Ahuja, J.S., 2006. Enhancing the design of web navigation systems: The influence of user disorientation on engagement and performance. Mis Q. 661–678.

Wu, Z., Jiang, Y., Liu, Y., Ma, X., 2020. Predicting and diagnosing user engagement with mobile ui animation via a data-driven approach. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. pp. 1–13.

Wu, Z., Kim, T., Li, Q., Ma, X., 2019. Understanding and modeling user-perceived brand personality from mobile application uis. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. pp. 1–12.

Wu, J., Swearngin, A., Zhang, X., Nichols, J., Bigham, J.P., 2023. Screen correspondence: Mapping interchangeable elements between UIs. arXiv preprint arXiv:2301.08372.

Wu, J., Zhang, X., Nichols, J., Bigham, J.P., 2021. Screen parsing: Towards reverse engineering of ui models from screenshots. In: The 34th Annual ACM Symposium on User Interface Software and Technology. pp. 470–483.

Xiang, W., Zhu, H., Lou, S., Chen, X., Pan, Z., Jin, Y., Chen, S., Sun, L., 2024. SimUser: Generating usability feedback by simulating various users interacting with mobile applications. In: Proceedings of the CHI Conference on Human Factors in Computing Systems. pp. 1–17.

Yan, A., Yang, Z., Zhu, W., Lin, K., Li, L., Wang, J., Yang, J., Zhong, Y., McAuley, J., Gao, J., et al., 2023. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. arXiv preprint arXiv:2311.07562.

Yang, Z., Liu, J., Han, Y., Chen, X., Huang, Z., Fu, B., Yu, G., 2023. Appagent: Multimodal agents as smartphone users. arXiv preprint arXiv:2312.13771.

Yang, B., Xing, Z., Xia, X., Chen, C., Ye, D., Li, S., 2021. Don't do that! hunting down visual design smells in complex uis against design guidelines. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering. ICSE, IEEE, pp. 761–772.

Zhang, X., De Greef, L., Swearngin, A., White, S., Murray, K., Yu, L., Shan, Q., Nichols, J., Wu, J., Fleizach, C., et al., 2021. Screen recognition: Creating accessibility metadata for mobile applications from pixels. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. pp. 1–15.

Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. arXiv preprint arXiv:2403.13372.