Identifying the Gap Between UX Practitioners' Work Practices and AI-Enabled Design Support Tools

YUWEN LU, University of Notre Dame, USA
CHENGZHI ZHANG, Carnegie Mellon University, USA
IRIS ZHANG, Kings Christian Collegiate, Canada
TOBY JIA-JUN LI, University of Notre Dame, USA

User interface (UI) and user experience (UX) design have become an inseparable part of today's tech industry. Recently, advancements in machine learning (ML) have opened up new opportunities for innovations in UI/UX design tools. However, many prototypes in this field haven't been adopted in practice and a gap between ML-enabled tools and designers' day-to-day work practices exists. To learn the underlying reasons and bridge this gap, we conducted contextual interviews with 8 UX professionals to understand their practices and identify opportunities for more translational research. We found that most current ML-enabled design tools focus on graphical interface elements, while activities involving more 'design thinking", such as needfinding, are as crucial for designers. Many existing system prototypes were designed for overly-simplistic scenarios that fail to integrate design projects' practical considerations. We also identified 4 areas in the UX workflow that can benefit from additional ML-enabled assistance: design inspiration search, design alternative exploration, design system customization, and design guideline violation check.

CCS Concepts: • **Human-centered computing → Interaction design process and methods**.

Additional Key Words and Phrases: User Experience (UX), Human-AI Collaboration, design-support tools, data-driven design

ACM Reference Format:
Yuwen Lu, Chengzhi Zhang, Iris Zhang, and Toby Jia-Jun Li. 2022. Identifying the Gap Between UX Practitioners' Work Practices and AI-Enabled Design Support Tools. In *Proceedings of the InContext: Futuring User-Experience Design Tools Workshop at CHI Conference on Human Factors in Computing Systems (CHI '22), April 30–May 6, 2022, New Orleans, LA.* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In the tech industry nowadays, UI and UX design is a key element in the life cycle of product development. Not only do user interfaces contribute to the aesthetics of a product, but they also serve as an indispensable part of user experience and convey a company's branding and style.

However, UI design and development are time-consuming and error-prone [16]. Many researchers have worked on building design support tools to improve user interface creators' work efficiency. In the 1990s, early research projects such as the SILK system [13] and Garnet [17] were conducted for this purpose. Later, with the growth of UI and UX design as an individual profession, many commercial design and prototyping tools including PhotoShop, Sketch, Webflow, and Figma are developed to support graphical UI and interaction design. These tools have greatly helped designers in creating interfaces and prototypes for different use cases, contexts, and devices and are adopted by a wide range of organizations worldwide.

Recently, advances in machine learning (ML) have enabled data-driven approaches to support UI/UX design [11]. The introduction of large-scale datasets such as RICO [5] laid the foundation for training deep neural networks for user-interface-related tasks. Many research projects in areas including design search [9], UI generation [28], and UI understanding [14, 25] have followed. However, many of the ML-based research projects' impact remained within the academic research community and haven't succeeded in making practical influences on industry practices [11]. This phenomenon is common across HCI research and has been identified as the "research-practice gap". This term refers to the fact that HCI research findings, supposedly helpful for UX work, are rarely utilized by UX practitioners in the industry [3, 18]. Bridging this gap requires translational research that identifies practitioners' specific needs and provides translational resources for them to benefit from the latest technical advances and academic research findings.

In this work, to bridge the research-practice gap for ML-enabled UI/UX design support tools, we conducted a study with 8 UX practitioners with varying experience and backgrounds to learn about their work practices. We also used existing ML-based design support prototypes in the form of storyboards to gather feedback, understand user needs, and solicit design ideas. Through qualitative analysis, we identified 4 opportunity areas for ML to facilitate designers' work: (1) design inspiration search; (2) design alternative exploration; (3) design system customization; and (4) design guideline violation check.

We identified several gaps between current research projects and designers' actual needs. Current design support tools using ML mostly focus on graphical interface elements, while the design activities that involve more "design thinking" and less graphical elements, such as brainstorming and needfinding, are more helpful for designers to create enjoyable and usable designs. In addition, existing models generate outputs that are too generic and not specific to designers' problems domains or their companies' design styles. Designers need to invest substantial efforts in customizing these generic outputs to fit their purposes, and such efforts are so great that many of our participants claimed these model outputs are barely helpful. We believe these issues come from the fact that most ML models in this area work in overly-simplified scenarios and fail to take many real-world design factors into account.

2 METHODOLOGY

2.1 Study Procedure

In this need-finding study, our goals were: (1) understanding UX practitioners' work practices and challenges, (2) getting practitioners' feedback on existing research prototypes using ML to facilitate UX work, and (3) identifying future design opportunities for ML-enabled design support tools. To fulfill these goals, we conducted contextual inquiry and speed dating with 8 UX practitioners. These participants were recruited through social media advertisement and through a snowball method [8]. A description of participants' demographics is shown in Table 1.

In each study session[1], we first asked questions regarding designers' work practices through one or two previous design project examples. Then, we used 4 speed-dating storyboards showing scenarios of using ML to facilitate their work practices to solicit their feedback. The speed dating session was followed by questions regarding their ideas and concerns for using ML to automate their workflows. Lastly, we finished the interview with questions regarding differences between UI and UX design to find opportunities beyond UI manipulation with the help of ML. All interviews

[1]The study protocol has been approved by the IRB at our institution.

Table 1. Demographics of Study Participants

ID	Gender	Education	Industry	Job Title	Yrs. of Experience	Company Employee Count
P1	Female	Master's	Healthcare	UX Designer	1-3	1,000 - 10,000
P2	Female	Master's	Info. Services	UX Designer	1-3	> 10,000
P3	Female	Master's	Info. Services	UX Researcher	3-5	> 10,000
P4	Female	Master's	Entertainment	UX Researcher	Less than 1	1,000 - 10,000
P5	Female	Bachelor's	Government	Program Lead	1-3	1,000 - 10,000
P6	Female	Master's	Info. Services	UX Designer	1-3	> 10,000
P7	Female	Master's	Info. Services	UX Designer	1-3	> 10,000
P8	Female	Bachelor's	Info. Services	UI/UX Designer	1-3	> 10,000

were conducted online and lasted around 60 minutes. Each participant was compensated $15 for their time. We recorded all interviews with the permission of our participants and used the tool Grain [1] to transcribe them for analysis.

2.2 Contextual Inquiry

Contextual inquiry has been widely used by the HCI community [23]. It emphasizes the context of studied behaviors and puts research participants in the context where they would normally use the systems. It is powerful for observing user behavior and uncovering their underlying rationales and mental models.

In our study, our questions about designers' work practices depend greatly on project contexts. Directly asking such questions without having participants refer to actual projects will lead to ambiguities. As a result, we decided to use contextual inquiry. The context we used in this project is mostly digital design documents. Specifically, we asked participants to go through some of their previous design project files to illustrate how they generated insights and made design decisions. The interview questions centered around concrete examples of insights and decisions in the previous projects, allowing participants to recall more details of their projects and provide more useful information.

2.3 Speed Dating

In addition to contextual inquiry, we conducted speed dating. Speed dating is a technique where researchers show users multiple possible design solutions and gather feedback [4]. Its usefulness lies in that after seeing many alternative solutions, users can have a better understanding of their true needs independent of the example solutions they looked at. This helps the interviewers to learn about users' true needs, which even users themselves often didn't realize.

In our study, we presented each participant with 4 storyboards of scenarios using ML-enabled design-support tools. A portion of a storyboard is displayed in Figure 1. When creating the storyboards, we selected some prototypes from previous literature that specifically targeted a process in UI/UX design such as [9, 10]. We also created 2 storyboards in explorations of opportunities to facilitate design work with ML that we have previously identified. After we presented each storyboard to the participant, we asked about their thoughts on each scenario's usefulness in their own daily work. We also followed up with questions on how the interviewee would change the scenarios when appropriate. Most importantly, speed dating can help users identify real needs that potentially lie outside of the given storyboard examples [4]. We followed this section with questions prompting our interviewees to think of additional areas they think ML can help with any potential concerns they would have.

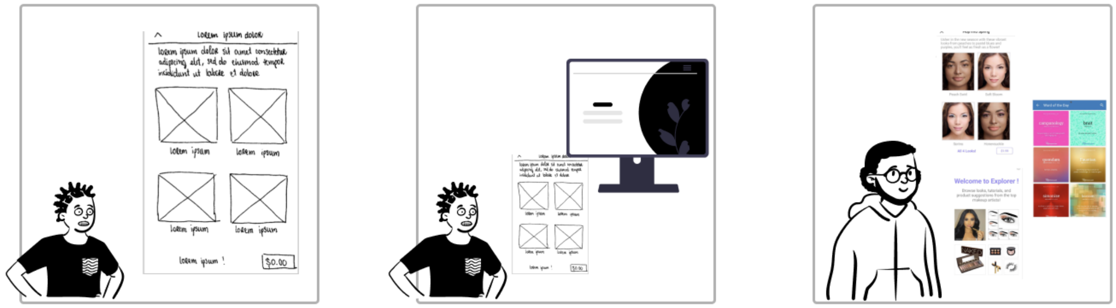

Fig. 1. An example storyboard for a sketch-based design example retrieval tool used during interviews.

2.4 Qualitative Analysis Methods

Two authors of the paper conducted a qualitative analysis of the interview transcripts using thematic analysis [22] and affinity diagramming [19]. The first round of analysis was open coding on interview transcripts using the tool Grain [1]. Two authors collaboratively went through the recordings, highlighted portions of the transcripts that are relevant to our research topic, and wrote a descriptive text for each highlight. The goal of this round is to identify relevant and valuable information in the transcripts. Then, two authors imported all of the open codes into Figma and conducted the second round of coding. The second round was the beginning of the inductive thematic analysis. During this round, two authors gathered open codes that are relevant to each other, formed clusters, and wrote a summary text for each cluster. Each summary text represented a specific idea discussed by our participants about detailed processes or issues they face in their design workflow. The third round of coding followed the second one as two authors grouped and summarized the previous summary texts. Each of the new, higher-level summary texts represents a design opportunity or a more general issue identified in our interview data.

3 FINDINGS

3.1 Design Opportunities

From our analysis, we were able to identify four areas in UI/UX designers’ work that can potentially be supported by ML. For some of these opportunities, there have already been researchers working on them; the limitations of existing work in these areas will be further discussed in this section.

3.1.1 Design Inspiration Search. UI/UX designers use many references to generate inspiration for their own design. The references are not limited to examples, also can be guidelines and best practices. During our interview, P2 mentioned that “experienced designers are experienced because they have all the examples and inspirations stored in their mind”. Similarly, P8 expressed that “as a designer, you need to have some patterns in your memory, but that requires experience”. Designers need references to see what patterns fit the current design context and need to weigh the pros and cons of each reference before combining them into a desired one. We learned that in practice, designers usually curate their own reference libraries to find inspirations for their own design. They usually search for reference examples similar to their design goals based on **functionalities, problem domains, and visual styles**.

However, designers often have difficulty finding many relevant examples with these metrics. Sometimes, designers do now know what keywords to use or are constrained by the limited number of keywords they come up with. At the same time, designers expressed the need for a large number of references to generate good ideas, which is in line with suggestions for designing creativity support tools from previous literature [21]. This creates the opportunity for facilitating design inspiration search with ML. A possible solution is to explore different modalities to search other than keywords. Designers want to find similar designs based on functionalities, problem domains, and visual styles. Previous work has utilized visual styles as a modality for searching. For example, Swire [9] enables designers to search for high-fidelity interface examples using hand sketches.

Nevertheless, little work has enabled search based on the app's problem domains and functionalities. Such new directions create novel technical challenges. We need to build ML models that can understand an app's problem domain or interface elements' functionalities. In addition, designing such apps requires a deeper understanding regarding different dimensions of similarity used for searching and the degree of similarity designer desire for the search results to be inspiring.

3.1.2 Design alternatives exploration. Designers usually look at different design possibilities and test out alternative solutions. In interviews, designers expressed interest in using ML to automatically generate alternatives for an existing design as exploration. P2 specifically pointed out that this will be more helpful for graphic elements such as color, layout, and font instead of more high-level ones such as ways to fulfill a user's need. More specifically, P6 mentioned that such exploration would be most helpful if the alternatives can be a bit "outside the box" and creative, to stimulate more creativity from the designer. They imagined adding a certain degree of randomness to the output result can help introduce this creativity when exploring alternatives. This calls for new ML model architectures that add an additional layer of randomness over the generated results, similar to variational autoencoders [12]. Designers also want to have control over the degrees and forms of randomness to create results that fit their purposes.

3.1.3 Design system customization. The introduction of design systems in many companies created opportunities and challenges at the same time. On one hand, they improve designers' and developers' efficiency by providing a library of standard visual components (e.g. buttons, forms, navigation bars) that conforms to the organizations' branding styles. Designers and developers can directly use them since most of these components have already been designed, programmed, and tested. However, on the other hand, there are occasions in which design systems actually reduce designers' efficiency. In our interviews, 5 out of 8 participants talked about scenarios where these standard components do not fit their specific design purposes and they have to customize or redesign them. This creates a burden for both designers and developers and undermines the potential outcomes the design could have achieved.

For customized elements, designers sometimes need to check other teams whether their redesign conforms to the company's style guidelines. Since these elements are new, developers also have to write code from scratch to implement them, which significantly increases their workload and developers are usually reluctant to do it. In these cases, interactive systems that automatically adapt customized widgets or components to the company's design styles and guidelines will be tremendously helpful. Possible solutions include borrowing concepts from image style transfer algorithms [7] and applying them to UI widgets. Regarding implementation code generation, existing research such as GUIS2Code [6] and Chen et al. [2] can already facilitate similar tasks after building the desired design.

3.1.4 Automatic design guideline violation check. During the interview, P2 expressed that some designers "are not paying too much attention to inclusiveness (e.g., accessibility), but it is very important". They proposed that if some ML systems can provide friendly reminders for "simple things like the color contrast, … keyboard navigation, and a lot of other details (related to accessibility)" it would be greatly helpful. There has been much research using ML to understand screen elements [14, 25]. By utilizing such models, it's possible to build applications that automatically detect violations of accessibility design guidelines and prompt designers to make improvements. Another potential direction is to model users with varying levels of ability, test the app with the agents to simulate user testing sessions, and apply ability-based design principles [26]. More importantly, such research opportunities are not limited to accessibility guidelines and can be expanded to universal usability guidelines [24] or other forms of design guidelines outside accessibility.

3.2 Gaps Between Existing Tools and Designers' Needs

Compared to using generated design results from existing ML models, designers prefer to get inspiration from existing apps and create their own designs for several reasons. Firstly, designers don't have control over the generation process, combined with the fact that the generative system didn't provide any rationale behind the result, designers do not trust the system's output. P6 explicitly mentioned "I don't have that trust in the system, so I would question why the system suggests this. I need several stems of solutions so I can…have that control to compare at least some of them". Also, designers expressed that existing model outputs, such as the sign-in mock-up pages from [10], tend to be too generic, thus cannot be easily adopted by designers to suit their own needs. As discussed in Section 3.1.3, designers spend more time adapting the initial design to company styles and guidelines than creating initial mock-ups, while current generative models can only support the latter. On the other hand, existing apps are more likely to have been through user tests and conform to best practices and guidelines. They can serve as better examples for designers.

Besides, current ML models are rarely helpful for design activities that do not involve graphical interface elements, the ones described by designers as involving more "design thinking". From an ML standpoint, interface elements are easier to manipulate due to the simplicity of their data representation; however, designers articulated that in UX work, those activities that involve more design thinking, e.g. user interviews, brainstorming sessions, and user testings, are more important for creating usable and enjoyable designs. During our interview, P6 mentioned that "for more complex features, the rationale behind designing something is more important than the visual elements and layouts". Coming up with such rationales requires a deep understanding of users' intentions and needs. It makes up a great portion of designers' daily jobs, however, most current ML-enabled UX design support tools overlooked it.

Also, existing ML models are not helpful in generating outputs that are context-specific to the designers' problem domain. For example, existing models only work best for generic interfaces that are common for many apps, e.g., sign in pages, card list pages, user profile pages. When a designer wants to design a list of all doctors available in an area for a healthcare app, existing ML models would not understand contexts such as the information to display for each doctor or the order to list the doctors. However, these are all important design decisions to be made by designers. We argue that current ML models usually work in an overly-simplified scenario and don't take many real-world parameters into consideration. This leads to exceedingly generic design solutions that require too much customization done on the designer's end. Some study participants argued that such customization effort is so much that the generic generated results are almost not helpful for them.

Moreover, one of the main tasks for UX professionals in their daily jobs is to convince other non-UX team members of their work's value and quality. This task is closely related to their design generation process since value and quality are usually communicated by illustrating designers' design decisions and the underlying rationales. However, current ML models act on a model to replace designers in generating design interfaces, instead of complementing designers' agency and creativity. If designers use ML-generated design results, it's largely impossible to justify a model's design with rationales because they are not much involved in the generation process. One possible solution to this issue is to include additional model inputs such as explicit design decisions and user insights to generative ML models. Also, in our interviews, P8 expressed that when an ML model generates a design, it would be helpful for the model to provide some supporting evidence, such as well-known app examples that adopted a similar design layout. By incorporating such evidence, designers as well as other team members could have more confidence in the generated results. Researchers can get inspiration from the area of explainable AI [15, 20], especially those investigating generative models [27], to build ML models that generate explainable UI design.

REFERENCES

[1] Mike Adams. 2022. Cure your Zoom amnesia with Grain 2.0. https://grain.co/blog/announcing-grain-2-0

[2] Chunyang Chen, Ting Su, Guozhu Meng, Zhenchang Xing, and Yang Liu. 2018. From ui design image to gui skeleton: a neural machine translator to bootstrap mobile gui implementation. In *Proceedings of the 40th International Conference on Software Engineering*. 665–676.

[3] Lucas Colusso, Cynthia L Bennett, Gary Hsieh, and Sean A Munson. 2017. Translational resources: Reducing the gap between academic research and HCI practice. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. 957–968.

[4] Scott Davidoff, Min Kyung Lee, Anind K Dey, and John Zimmerman. 2007. Rapidly exploring application design through speed dating. In *International Conference on Ubiquitous Computing*. Springer, 429–446.

[5] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 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*. 845–854.

[6] Zhen Feng, Jiaqi Fang, Bo Cai, and Yingtao Zhang. 2021. GUIS2Code: A Computer Vision Tool to Generate Code Automatically from Graphical User Interface Sketches. In *International Conference on Artificial Neural Networks*. Springer, 53–65.

[7] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015).

[8] Leo A Goodman. 1961. Snowball sampling. *The annals of mathematical statistics* (1961), 148–170.

[9] Forrest Huang, John F Canny, and Jeffrey Nichols. 2019. Swire: Sketch-based user interface retrieval. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–10.

[10] Forrest Huang, Gang Li, Xin Zhou, John F Canny, and Yang Li. 2021. Creating User Interface Mock-ups from High-Level Text Descriptions with Deep-Learning Models. *arXiv preprint arXiv:2110.07775* (2021).

[11] Yue Jiang, Yuwen Lu, Jeffrey Nichols, Wolfgang Stuerzlinger, Chun Yu, Christof Lutteroth, Yang Li, Ranjitha Kumar, and Toby Jia-Jun Li. 2022. Computational Approaches for Understanding, Generating, and Adapting User Interfaces. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA.

[12] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[13] James A Landay. 1996. SILK: sketching interfaces like krazy. In *Conference companion on Human factors in computing systems*. 398–399.

[14] Toby Jia-Jun Li, Lindsay Popowski, Tom Mitchell, and Brad A Myers. 2021. Screen2Vec: Semantic Embedding of GUI Screens and GUI Components. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.

[15] Q Vera Liao, Moninder Singh, Yunfeng Zhang, and Rachel Bellamy. 2021. Introduction to explainable ai. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–3.

[16] Brad Myers, Scott E Hudson, and Randy Pausch. 2000. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7, 1 (2000), 3–28.

[17] Brad A Myers, Dario A Giuse, Roger B Dannenberg, Brad Vander Zanden, David S Kosbie, Edward Pervin, Andrew Mickish, and Philippe Marchal. 1995. GARNET comprehensive support for graphical, highly interactive user interfaces. In *Readings in Human–Computer Interaction*. Elsevier, 357–371.

[18] Donald A Norman. 2010. The research-Practice Gap: The need for translational developers. *interactions* 17, 4 (2010), 9–12.

[19] Kara Pernice. 2018. Affinity Diagramming: Collaboratively Sort UX Findings & Design Ideas. https://www.nngroup.com/articles/affinity-diagram/

[20] P Jonathon Phillips, Carina A Hahn, Peter C Fontana, David A Broniatowski, and Mark A Przybocki. 2020. Four principles of explainable artificial intelligence. *Gaithersburg, Maryland* (2020).

[21] Mitchel Resnick, Brad Myers, Kumiyo Nakakoji, Ben Shneiderman, Randy Pausch, Ted Selker, and Mike Eisenberg. 2005. Design principles for tools to support creative thinking. (2005).

[22] Maria Rosala. 2019. https://www.nngroup.com/articles/thematic-analysis/

[23] Kim Salazar. 2020. https://www.nngroup.com/articles/contextual-inquiry/

[24] Ben Shneiderman, Michael Leavitt, et al. 2006. Research-based web design & usability guidelines. *Washington DC, Department of Health and Human Services* (2006).

[25] Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. 2021. Screen2Words: Automatic Mobile UI Summarization with Multimodal Learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 498–510.

[26] Jacob O Wobbrock, Shaun K Kane, Krzysztof Z Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing (TACCESS)* 3, 3 (2011), 1–27.

[27] ZiCheng Zhang, CongYing Han, and TianDe Guo. 2021. ExSinGAN: Learning an Explainable Generative Model from a Single Image. *arXiv preprint arXiv:2105.07350* (2021).

[28] Tianming Zhao, Chunyang Chen, Yuanning Liu, and Xiaodong Zhu. 2021. GUIGAN: Learning to Generate GUI Designs Using Generative Adversarial Networks. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 748–760.