

Revisiting Piggyback Prototyping: Examining Benefits and Tradeoffs in Extending Existing Social Computing Systems

DANIEL A. EPSTEIN, University of California, Irvine, USA

FANNIE LIU, Snap Inc., USA

ANDRÉS MONROY-HERNÁNDEZ, Snap Inc. & Princeton University, USA

DENNIS WANG, University of California, Irvine, USA

The CSCW community has a history of designing, implementing, and evaluating novel social interactions in technology, but the process requires significant technical effort for uncertain value. We discuss the opportunities and applications of “piggyback prototyping” [37], building and evaluating new ideas for social computing on top of existing ones, expanding on its potential to contribute design recommendations. Drawing on about 50 papers which use the method, we critically examine the intellectual and technical benefits it provides, such as ecological validity and leveraging well-tested features, as well as research-product and ethical tensions it imposes, such as limits to customization and violation of participant privacy. We discuss considerations for future researchers deciding whether to use piggyback prototyping and point to new research agendas which can reduce the burden of implementing the method.

CCS Concepts: • Human-centered computing → Social networking sites.

Additional Key Words and Phrases: social computing, systems, piggyback prototyping, social media

ACM Reference Format:

Daniel A. Epstein, Fannie Liu, Andrés Monroy-Hernández, and Dennis Wang. 2022. Revisiting Piggyback Prototyping: Examining Benefits and Tradeoffs in Extending Existing Social Computing Systems. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW2, Article 456 (November 2022), 28 pages. <https://doi.org/10.1145/3555557>

1 INTRODUCTION

The CSCW research community has a long history of designing, implementing, and evaluating novel social communication technologies to explore how to address the limitations of existing systems or envision alternative futures. Through this work, we as a community aim to address classic CSCW problems like connecting people with similar interests and goals, helping them have conversations about important topics, minimizing harmful content, and more. As outlined in the joint UIST+CSCW panel at the 2020 conference [5], the decades of systems research on collaborative and social computing have led to a richer understanding of how to design these systems and important improvements in the social tools we use daily.

It is well known that there are significant barriers to designing, implementing, and evaluating new systems and design ideas to answer research questions about social communication. For example, Bernstein et al. remark on how challenging it is to build up a social computing system, and how research incentives are often at odds with commercial incentives to do so [4]. Community events

Authors' addresses: Daniel A. Epstein, epstein@ics.uci.edu, University of California, Irvine, USA; Fannie Liu, Snap Inc., USA, fannie@snap.com; Andrés Monroy-Hernández, Snap Inc. & Princeton University, USA, amh@snap.com; Dennis Wang, dennisw7@uci.edu, University of California, Irvine, USA.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2022 Copyright held by the owner/author(s).

2573-0142/2022/11-ART456

<https://doi.org/10.1145/3555557>

often bring together researchers to discuss these issues and develop paths forward, like the Social Computing Systems Camp [108], the CSCW 2021 panel discussion on the research community's role in reforming or remaking social technologies [98], and the yearly CrowdCamp workshop at HCOMP [3].

To address some of these concerns, Grevet & Gilbert propose *piggyback prototyping*, or building new ideas for social computing systems on top of existing systems, as one technique to more rapidly support evaluating design ideas prior to developing an entirely new social computing system [37]. In the years since this proposal, social computing platforms have increasingly incorporated resources and APIs which allow for the customization and modification needed to piggyback prototyping. For example, programmable and configurable bots are central to chat and work platforms like Slack, Discord, and WeChat [58, 61], online communities like Reddit and Twitch [48, 90], and peer production communities like Wikipedia [33]. Social computing systems also often allow people to share content they generate in other apps and tools, such as through share sheets provided by phone operating systems for sharing to social media or chat apps (e.g., Twitter, Instagram, iMessage, Google Chat) and more sophisticated APIs like Snap's Creative Kit and Lens Studio [94, 95]. Researchers often leverage these resources when implementing piggyback prototypes to answer their research questions, such as creating research apps that can post to participant's social media accounts to evaluate new kinds of content [26, 77, 96] or creating bots that can post messages to feeds where participants interact with one another to evaluate new content summarization or organization strategies [88, 91, 106].

With the benefit of hindsight, we revisit and expand on Grevet & Gilbert's work by critically examining and discussing the benefits and tradeoffs of research which uses the method of designing, developing, and evaluating social communication experiences that piggyback off of existing systems in comparison to building them from scratch. We summarize and draw on observations from researchers who developed about 50 prior social computing systems to answer their research questions, using three case studies where we are more familiar with their design and evaluation to further describe how the method can support different research goals. While Grevet & Gilbert suggest that piggyback prototyping can help assess the critical mass of interest prior to spending resources developing a new social computing platform, we surface that the research community has additionally used piggyback prototyping to answer research questions about the design of social computing systems. For example, piggyback prototypes have contributed to different onboarding and social support interventions for introducing new members to the norms of established social communities like Wikis and Q&A sites [15, 31, 69, 81]. Research systems have also piggybacked to explore how designs can support sharing new kinds of information in feed-based conversations, like health data [26, 77, 78] and biosignals [18, 64, 68].

Drawing on the research literature's frequent use of piggyback prototyping to create recommendations for the design of social computing systems, we describe and discuss the benefits of the method and the tensions faced compared to developing social computing systems from scratch. We identify benefits which are *intellectual*, such as improved ecological validity and the ability to study more complex social relationships. We also address *technical* benefits, such as the ability to leverage existing well-tested features. At the same time, we characterize *research-product tensions* such as the increased expectation of polish of all aspects of the system and limits to customization, as well as *ethical tensions* around access to research data outside of the research team, violation of participant privacy, and exclusionary practices reinforced by piggybacking on platforms. In summarizing these benefits and tradeoffs, we advocate for more CSCW systems research to consider piggybacking as a method and for more CSCW systems to enable being piggybacked on by later researchers.

By critically examining piggyback prototyping systems, we contribute:

- A review of design challenges for social computing systems where piggyback prototyping has been used to help address, extending Grevet & Gilbert's focus on evaluating interest level [37]. We draw on about 50 past systems which have leveraged the method, characterizing five design challenges: creation and personalization of social content, ordering and filtering of content, establishing affinity groups, onboarding new members, and content moderation.
- Descriptions of core benefits and tensions of piggyback prototyping. We characterize four intellectual and technical benefits to evaluating design ideas for social computing systems within existing systems: in-situ evaluation, asymmetric participation and audience access, leveraging built-in mechanics, and simplified and organized participant data collection. We also articulate two main tensions around using the method: tensions between the goals of the research and the goals of the product being piggybacked off of, and ethical tensions around using existing platforms for research. In reflecting on three research projects, SnapPI, Everyday Biosignals, and Mindful Garden, we consider how piggyback prototyping improved aspects of these works while also introducing design and evaluation challenges.
- Recommendations for future researchers and platform creators interested in using or supporting piggyback prototyping. We describe three considerations for researchers deciding whether to use the method: whether platforms allow for the desired customization, whether people are already using the platform to communicate, and whether piggybacking introduces privacy or ethical concerns. We advocate for piggyback prototyping by building on prior arguments around the difficulties of CSCW systems research, and point to new research agendas, such as developing customizable authoring tools for existing social platforms.

2 REDEFINING PIGGYBACK PROTOTYPING SOCIAL COMPUTING EXPERIENCES

Grevet & Gilbert define *piggyback prototyping* as a “social computing system prototyping technique that utilizes existing social platforms to evaluate novel social interactions for large-scale systems” [37]. In describing stages of piggyback prototyping, Grevet & Gilbert focus on using the method to evaluate whether an idea for a new community is likely to attract a critical mass of interested people prior to investing development effort. We broaden this definition to include *prototyping a novel social interaction utilizing an existing platform, where the intent is to address a common communication problem, barrier, or challenge or introduces a new communication capability, opportunity, or vision*. In addition to measuring interest in an idea prior to scaling up development, our definition includes prototyping with the goal of creating recommendations for improving social computing systems which either already exist or could exist in the future. The recommendations could aim to improve a specific system or platform (e.g., Facebook, SMS) or systems that have shared characteristics (e.g., systems with ephemeral messaging, systems where people have large audiences).

For this paper, we focus on the potential for piggyback prototyping to support research questions around improving a person’s *social or communication practices* in a system, as understanding and supporting socialization and communication is a fundamental research goal of the CSCW community. This distinction leaves many social computing systems that have built on existing systems outside of the scope of our discussion. For example, research tools have often piggybacked off of social computing systems to promote other goals, such as embedding microtasks into social feeds for productivity [43] or temporarily blocking or altering social websites to encourage a person to do other activities [1, 49, 54, 72]. The interventions of these systems may have a side-effect of potentially reducing how frequently people use that system to communicate, but modifying social interaction was not the primary intent. Similarly, research systems in Crowdsourcing have piggybacked off of existing systems (e.g., Amazon Mechanical Turk, Upwork) to evaluate different workflows or structures [6, 52, 87], but these systems, by in large, have not modified how workers

communicate with one another. These examples highlight that piggyback prototyping can support a wide range of research goals beyond those we discuss.

CSCW research also frequently examines the impact of changes that social platform designers make on people's experiences with the platform. For example, Dasgupta & Hill used a policy change in the Scratch community to causally evaluate the impact of enabling cloud variables on learners' sharing behavior [20]. Evaluations are occasionally planned experiments around design features organized by the platform, such as Facebook's social contagion study [55]. The benefits and challenges of evaluating novel social computing systems has overlap with experimentation that the systems' creators or maintainers may perform, such as A/B testing. However, most of the implications we discuss in this work relate to designs created and evaluated by researchers outside the platforms, and who may have limited ability to modify the platform through available APIs or have partnerships with platform development teams.

Many systems research contributions, including social computing systems, do not necessitate eliciting formal feedback from the people for whom the technology is designed. Proof-by-existence, articulating a vision, or improving technical performance may all be more appropriate evaluations for many research contributions [36, 85]. Many of the benefits and challenges we articulate for piggyback prototyping are specific to evaluating the prototype in-situ with the population it was designed to support, such as through a deployment or a laboratory study. We also highlight that towards many research questions, understanding people's current social communication practices on an existing platform may be sufficient or more valuable than designing a piggybacked system that tests a different communication practice. We do not advocate for piggyback prototyping as a replacement for these sorts of contributions.

3 RESEARCH APPROACH

Building an understanding of the design challenges piggyback prototyping has been used to address, and the benefits and tradeoffs of the method, requires identifying a set of literature that contributes piggyback prototyped systems. To the best of our knowledge, there are no formal lists of literature or common keywords used in the literature to describe piggyback prototyping, removing the option of conducting a systematic literature review. At the time of writing, searching for "piggyback prototyping" and "piggyback prototype" in the ACM digital library resulted in 3 unique articles. Although we have drawn significant inspiration from Grevet & Gilbert's characterization [37], relatively few systems that have since built on existing social computing systems have cited the article. Similarly, searching for the names of commercial systems like Snapchat or Reddit would result in significantly more articles that do not meet the inclusion criteria than those that do, such as the study of people's use of the platforms.

To identify the literature that used the method, we, therefore, began with a list of about 20 articles we were aware of from our own experience conducting research in the space. Over the course of a year, we added to this list by (1) reading the literature that cite and are cited by those 20 articles (and looking at subsequent references), (2) examining the publication histories of the scholars who wrote the original articles, (3) perusing the recent and upcoming proceedings of prominent venues like CSCW, CHI, and UIST, and (4) asking our social networks. Through this process, we also refined our definition of piggyback prototyping of social computing experiences. In total, we identified about 50 articles which contributed piggyback prototyped systems. We identified piggybacking systems from as early as 2004, with most having been published in the last decade.

We do not claim that this list is complete, but rather that the benefits and challenges discussed by those articles reflect that of the method applied generally. Within and outside of CSCW, there is inevitably a great deal of research that piggybacks off of commercial platforms like Google Docs or Instant Messaging clients. At scale, it can be difficult to differentiate research systems which

aim to improve social and communication practices from those which aim to support other goals. While conducting a more thorough review is outside the scope of this research, critically examining how a particular platform or class of platforms (e.g., Google Docs or collaboration tools) have been extended to provide different social experiences would generate useful guidelines for toolkits looking to reduce the threshold to creating these sorts of systems.

To understand researchers' perceptions of piggyback prototyping, we analyzed the system description, method, discussion, and limitations sections of the articles we identified. We specifically looked for passages which described: (1) what intended benefits influenced the choice to develop a piggyback prototype rather than design a system from scratch, and (2) what limitations, downsides, or concerns they encountered with the method, if any. In total, we identified about 100 passages which articulated benefits or limitations, ranging from a few sentences to multiple paragraphs.

Following a thematic analysis [10] approach, the first author began by revisiting the passages and open coding each. The 15 codes included “*benefits of being a ‘real world’ system*”, “*needing to fit with a sharer’s platform preferences*”, and “*consideration of terms of service*”. Through multiple discussions as a research team, we then consolidated our codes and grouped these passages, naming and categorizing the benefits and tensions identified in prior research. Table 1 contains the full list of articles we identified, summarizing the existing systems that we commonly saw piggybacked on.

Table 1. Examples of existing social systems on which the research literature has piggybacked from our literature review.

Existing system	Exemplar piggyback prototype(s)
Facebook	Asynchronous Remote Communities (ARCs) [73, 74, 86], CommitToSteps [78], FeedVis [28, 29], Food4Thought [25], GoalPost [77], HeartLink [18], Regroup [2], Three Good Things [79], Yarn [26]
SMS, Chat, and IM Apps (e.g., WhatsApp, WeChat, KakaoTalk)	Chorus [47], DearBoard [38], Everyday Biosignals [64], Mental Health Chatbot [59], ReactionBot [68], TableChat [71], MyButler [13], WeRun [60], Whatfutures [56, 57]
Wikis and Forums	ForumReader [100], MoodBar [15], ORES [44], Snuggle [45], The Wikipedia Adventure [81], Wikipedia Teahouse [76]
Email	Bluemail [102], Calendar.help [17], Squadbox [75], Themail [21]
Streaming Platforms (e.g., Twitch, YouTube Live, DouYu)	BabyBot [91], Snapstream [105], StreamSketch [70], StreamWiki [69], VisPoll [14]
Slack	Hive [88], T-Cal [32], Tilda [106]
Twitter	@BabySteps [96], Airport Check-ins [37], WeMeddle [34]
Snapchat	Mindful Garden [67], Project IRL [19], SnapPI [101]
Reddit	CrossMod [11], PolicyKit [107]
Google Docs	Guided Chat [84], WearWrite [82]
YouTube	OtherTube [8]
Stack Overflow	Help Room [31]
Figma	Winder [51]
JCow	PhamilySpace [89]

4 DESIGN CHALLENGES PIGGYBACK PROTOTYPING HAS BEEN USED TO ADDRESS

To show how piggyback prototyping has utility beyond assessing the level of interest in a social computing system, we characterize the design challenges and visions that prior research has

leveraged the method to produce. To inform scholars considering the method, we further describe its benefits and challenges relative to developing social computing systems from scratch.

We identified five challenges in the design of social computing systems where researchers have leveraged piggyback prototyping to develop recommendations. These challenges do not necessarily articulate all design challenges where researchers use piggyback prototyping, but characterize some common CSCW challenges where the method is useful. In outlining these challenges, we expand the method's applications beyond Grevet & Gilbert's original intentions for it to evaluate interest in a social computing system [37].

4.1 Supporting Creation and Personalization of Social Content

Researchers often extend social computing systems to enable the sharing of new or rarely shared content, evaluating people's perspectives of that content. For example, research systems have supported sharing sensed data to social platforms, including biosignals like heart rate [18, 64], visual expressions as sensed by a webcam [68], and steps or other forms of physical activity [26, 77, 78]. While it may be possible for people to share this content with enough effort (e.g., by copying text or taking a screenshot of another app), systems often aim to lower the barrier to creating this content. As another example, Calendar.help sought to reduce the burden of creating meeting invitations that work for multiple people's schedules, offloading scheduling efforts over email to a bot [17]. While scheduling meetings and sharing invitations is widely done without Calendar.help, the system aimed to lower the burden.

Researchers also often extend social computing systems to examine how designs can encourage people to share content which is already commonly shared. For example, @BabySteps prompted people to share about their newborn's development on social networking sites to more readily engage people in platforms they are already using [96]. Three Good Things leveraged a positive psychology intervention, encouraging sharing positive behaviors on Facebook with the hope of creating stronger bonds among close ties [79].

Other systems have supported personalizing or customizing how content appears or is delivered. DearBoard enabled people to personalize keyboards in messaging apps to promote connectedness [38]. MyButler enabled scheduling timing of messages in KakaoTalk, allowing people to adjust when they shared and received messages [13].

4.2 Ordering, Filtering, and Summarizing Content

CSCW systems often aim to improve how social content is ordered or summarized. Systems like Tilda [106] and T-Cal [32] enabled tagging and summarizing group chats, evaluating their effectiveness by studying past and present group conversations on Slack. Similarly, systems like ForumReader and Themail aimed to improve forum organization and email exchanges to allow people to better understand the content and social dynamics [21, 100].

Other systems have modified algorithms underlying how social content is ordered or filtered, such as Facebook's social contagion study [55]. FeedVis similarly surfaced what content is shown and hidden in NewsFeed algorithms, enabling participants to inspect the algorithms' influence on the posts they received [28, 29].

4.3 Establishing or Evaluating Affinity Groups

Other research has leveraged existing social platforms to evaluate strategies for bringing people together around a shared topic or interest. For example, the Asynchronous Remote Community (ARC) method convenes groups on online platforms such as Facebook to participate in design workshops, contributing recommendations for systems looking to foster discussion on personal or sensitive topics like pregnancy and HIV [73, 74, 86]. Lambton-Howard similarly used WhatsApp

to bring together a peer support community around health after learning how central WhatsApp was to other forms of communication [57]. Other systems have piggybacked to evaluate strategies for promoting competition on a shared topic, such as Food4Thought incorporating daily food challenges into a Facebook group to promote food mindfulness [25].

Systems have also extended platforms for small group communication to steer conversations towards particular topics. For example, TableChat and WeRun both extended WeChat to encourage family conversations around healthy eating [71] and physical activity [60] through sharing data on those topics (e.g., food journal photos, step logs). In addition to contributing design strategies for having conversations, evaluation of these piggyback prototypes also helps contribute understanding how people use digital systems to communicate around these topics.

Researchers have also built on existing social platforms to explore approaches to changing the formation of the groups people chat and share in. For example, the Slackbot in Hive reconfigured groups of designers to better support collective design [88]. Systems such as ReGroup and WeMeddle supported people in creating social circles from among their existing friends on Facebook and Twitter [2, 34].

4.4 Onboarding New Members

CSCW research, particularly in the space of peer production, has often prototyped approaches to address the challenge of introducing and supporting more diverse people to online communities. For example, Teahouse, embedded into Wikipedia as an alternative to the established help system, introduced a mentorship system to help new editors receive more support [76]. The Wikipedia Adventure gamified learning editing skills towards the same goal, recruiting participants from commenters on active Wikipedia pages [81]. Other work has aimed to introduce skills, behaviors, or practices to new members, such as MoodBar looking to increase socialization [15] and Snuggle normalizing critique practices [45]. Outside of peer production, systems have aimed to help new users understand and follow community norms, such as Help Bar for formulating questions on Stack Overflow [31].

4.5 Content Moderation and Quality Control

In large online communities that generate a lot of content, moderating content has emerged as an area where increased system support can help manage the workload. For example, tools like CrossMod and ORES have leveraged machine learning to initially evaluate content from existing online communities like Subreddits and Wikis [11, 44]. On a smaller scale, Squadbox explores asking a person's friends to evaluate email content they receive to avoid online harassment [75]. Evaluation techniques for these content moderation tools have varied, including recruiting participants to use the tool over a short duration [75], mirroring posted content in a separate instance with the tool to get feedback from moderators [11], and publicly releasing the tool and studying its use [44].

5 BENEFITS OF PIGGYBACK PROTOTYPING FOR RESEARCHERS

Piggybacking off of existing social computing systems offers substantial benefits for researchers, including promoting more realistic evaluations, supporting varied study participation arrangements, and lowering the burdens of some aspects of development. We note that some of the benefits provide intellectual advantages like increased ecological validity, while others provide technical advantages like simplifying the development or deployment of a prototype.

5.1 Intellectual benefits

Piggyback prototyping enables answering research questions that would be more challenging or impossible to answer from deploying standalone systems, improving researchers' ability to understand how people use or perceive social tools in their everyday lives.

5.1.1 In-situ evaluation. Systems leveraging piggyback prototyping extend the overall benefits of field studies relative to other methods, such as helping avoid ecological gaps in understanding how people use technology in everyday life [92, 99]. In social computing systems, people often have well-defined practices around what app(s) they use to communicate with others. Griggio et al. argue that when deploying a new communication tool, not requiring participants to relocate their conversations can allow for closer observation of the circumstances participants do and do not value the tool [38]. Cho et al. similarly state that requiring participants to switch platforms would “inevitably bring a new factor to the user’s messaging experience” [13].

Evaluating in existing social contexts can also help participants more critically consider whether the extension helps address the communication challenge it was intended to address. For example, the evaluation of StreamWiki within an existing streaming platform (DouYu) pointed to a need for systems to better interact with other technologies people use for editing videos and communicating with their streaming audiences [69]. Similarly, participants using systems like GoalPost, Yarn, and CommitToSteps were faced with considering whether their audiences (in these cases, on Facebook) would be interested in that content [26, 77, 78]. It may be possible to derive parts of these insights from formative interviews or speculative studies, but deciding whether a proposed approach aligns with or improves existing communication practices deepens this understanding.

Integrating with existing platforms that participants are already using can help minimize the gap between who they communicate with during and outside the study. In deciding to piggyback off of certain systems, researchers frequently pointed to the market share of those systems or participants’ level of familiarity as important factors [13, 56, 59, 68, 69, 71]. For example, TableChat chose to piggyback off of WeChat because an overwhelming majority of formative study participants already used WeChat for communicating with family members [71]. Similarly, some iterations of the ARC method leveraged Facebook because people already formed groups on the platform for discussing the relevant health topics, leading to similar experiences in- and out-of study [73].

Building off of an existing system can also serve as a reminder for study participation, adherence, or commitment to behavior change. For example, Food4Thought integrated with a private Facebook group where study participants could communicate with one another [25]. Comments from other participants would often show up in their Facebook feed, reminding them to engage as they used the platform for other purposes. Three Good Things similarly hoped that participants’ friends might remind them to engage with the intervention if they noticed that they were not posting on Facebook [79].

5.1.2 Asymmetric participation and audience access. Leveraging existing social computing systems enable studies where participants can be involved with different levels of engagement, and even engage with non-participants. This practice is often beneficial for systems that support creation of new types of content, as the person generating the content can then share it on the existing social platform. Studies that do not leverage existing social networking sites often instead require participants sign up for the study in pairs or groups, which results in limitations around group structure [50, 103]. Some studies also note the limitation of not capturing participants’ typical communication practices, highlighting that participants wished they could use the standalone system to communicate with people who were not part of the study [65, 103]. Asymmetric engagement

also allows for snowball sampling, wherein an audience member can notice content shared by a study participant and decide to enroll in the study.

For some system designs, the novel components being evaluated primarily impact only one person, allowing for others using the social system to be unaware of the study. For example, the Bluemail email client contributed new approaches for email searching and categorizing, requiring little need for engagement with the people with whom participants were sending or receiving emails [102]. Similarly, when FeedMe introduced a new technique for directed content sharing, it allowed recipients to receive shared content over emails they already used rather than sign up for the service themselves [7]. Conversely, other systems have benefitted from reporting on the perspectives of multiple stakeholders. For example, in evaluating the Yarn system for structured storytelling on social media, Epstein et al. asked field study participants to disclose who they shared the content with, and sent those audience members a short survey on their experience receiving the content [26].

5.2 Technical benefits

As suggested by Grevet & Gilbert piggyback prototyping avoids needing to reinvent features already present in existing social systems, such as social mechanics and participants' conversation logs. We further note benefits to the research, like participant familiarity with existing mechanics and simpler analysis.

5.2.1 Leveraging built-in mechanics. Compared to systems intended for individual use, systems with communal and collective mechanics often require significant development to ensure persistence of information across multiple people, features for socially engaging with one another (e.g., messages, likes, comments), methods for creating and customizing content, and more. Leveraging existing versions of these features can allow the development effort to focus on the interactive components most relevant to the research questions. Researchers often mentioned the benefit of a lower development and management burden for the research team, versus creating a social system from scratch [37, 73, 89]. Importantly, this benefit can lead to faster iteration on design ideas and evaluation of them, allowing the research community at large to more rapidly design, evaluate, and share ideas for improving social computing systems [23].

Leveraging mechanics from existing systems also promotes familiarity, both in overall interaction with a social computing system and specific features. For example, Tilda [106] leveraged the social mechanisms of custom slash commands and emoji reactions on Slack, while Three Good Things [79] leveraged liking and commenting built into Facebook. PhamilySpace highlighted that building on an existing system ensured participant familiarity with the technology, allowing the evaluation to focus on social interactions rather than platform design choices [89]. Novelty effects are also common in system deployment, where interest and engagement are high at the beginning of the study and taper over time. Minimizing new mechanics can possibly reduce this effect, as the design introduces less novelty overall.

Building off of existing systems also enables participants to continue using their typical communication strategies alongside the new strategy being evaluated. For example, in the UI design tool Winder, Kim et al. note that participants valued being able to use their typical chat alongside the new communication channel Winder introduced, leveraging the benefits of both [51].

5.2.2 Simplified and Organized Participant Data Collection. Particularly for public and semi-public social systems like some social media, piggybacked systems frequently touted the ease of collecting participant data. For example, in leveraging Twitter, Grevet & Gilbert note the value of being able to log participant response or lack of response to their intervention via Twitter's APIs [37]. Other studies similarly describe the benefit of analyzing timestamped and linked Facebook messages

from participants and audience members [18] or full chat logs from Twitch [91]. Within the ARC Facebook groups, Macleod et al. comment that the sheer volume of participant data (comments, likes, timing of messages) provides rich insight into participant experiences, but can even make the analysis overwhelming without organization and structure [73].

6 TENSIONS FOR RESEARCHERS WHEN PIGGYBACK PROTOTYPING

Though piggyback prototyping has significant benefits, utilizing the method also poses difficulties for researchers. Many challenges are around the tension between the nature of the piggybacked system as a product, as well as ethical questions resulting from leveraging an existing system.

6.1 Research-product tensions

Most of the systems that researchers piggyback off of are commercial applications, concerned with addressing the needs of their wide user base over the needs of researchers. Study participants using piggyback prototypes may extend their usability and performance expectations for commercial applications to the prototypes, increasing the need for them to be polished. Researchers also often wish to customize commercial tools in ways that are understandably not supported through APIs, limiting what piggyback prototypes they can create.

6.1.1 Expectation of polish. Because participants use piggybacked social computing systems to share content with non-participants, the expectations around system polish are often higher. For example, participants have expressed a need for many customization options to avoid repetitiveness [26], or critiqued the visual appearance of the content generated by the research system [78]. While framing the research study goals and setting expectations for participants may allay some of these concerns, they inevitably impact participants' use and impressions of the system and the ability to evaluate the effectiveness of design approaches.

Leveraging systems where people are already communicating further requires care to ensure that the intervention does not interfere with participant's other communication. For example, in Squadbox, Mahar et al. argued that if anything were to go wrong (e.g., emails did not get delivered), it would negatively impact their participants [75]. They thus had to test their system extensively, because losses would not only impact their research but the well-being or productivity of their participants or others they communicate with. When designing and evaluating BabyBot, a chatbot for Twitch that learns from how others converse, Seering et al. had to anticipate what problematic phrases people might attempt to make the bot learn, and develop strategies to moderate and mitigate that risk to avoid subjecting audience members to inflammatory remarks [90].

Piggyback prototypes often build on top of a single social platform to reduce the development burden. However, people often have idiosyncratic communication practices, defining rules for what app they use to communicate with whom and about what [83]. Participants using piggyback prototypes may find that the piggybacked platform does not align with the communication channel they prefer to use for that content [7] or with other tools in their communication ecosystem [69].

6.1.2 Limits to customization. Many piggyback prototypes rely on clever workarounds or hacks to implement the social mechanisms they aim to evaluate. For example, some systems require disabling and replacing features of the systems they are prototyping on, such as disabling KakaoTalk's native notifications to allow MyButler to provide its own notifications instead [13]. Others require installing additional components developed by neither the researchers nor the piggybacked platform, such as a scheduling chatbot in TableChat [71]. These workarounds also occasionally exploit potential privacy or security vulnerabilities with the piggybacked social systems or use their features in unintended ways. For example, DearBoard required taking a screenshot of a chat between two people in order to customize the keyboard for that interaction [38]. Although all screenshots

remained on the device, information about a person's conversation was still removed from the social interaction in order to implement the custom keyboard.

Technical barriers to customizing existing social tools also limit the kinds of interactions which can be supported. For example, in summarizing conversations on Slack with Tilda with a bot, Zhang et al. note that a more ideal interaction would introduce an overlay on top of or alongside existing conversation, but neither was possible [106]. As a result, summaries took up significant space in chat streams, which they worried would be distracting for participants. Relatedly, in @BabySteps, Suh et al. point out that the character limit imposed by Twitter would limit prompts for participants to report on certain baby milestones [96]. As a result, they had to adjust the wording of certain milestones, which could have impacted the reliability of the prompts.

Platforms often have policies which influence what kinds of interventions can be piggybacked. For example, when Ford et al. piggybacked Help Room on Stack Overflow, the platform's chat feature was only accessible to people who had sufficient reputation on the platform (e.g., 20 reputation points), interfering with the intervention's intent to help novices [31]. ReGroup notes specific evaluation choices required to stay within Facebook's Terms of Service [2]. Other research systems point to technical limitations of APIs for piggybacking which may negatively impact participants' experience with the system, such as rate limits [29, 34, 75] and unnecessary or distracting messages generated by the system being piggybacked on top of [13, 17].

Platform limitations may also impact the choice of platform which can be piggybacked. In PolicyKit, available API endpoints limit what social platforms the tool could piggyback on top of [107]. Zhang et al. remarked that Facebook Groups did not allow reverting or deleting messages, and changes were necessary to account for different audience response structures (e.g., emoji in Slack and Discord, up or downvotes on Reddit). Focusing on a single platform may limit generalizability of the design strategies proposed by a prototype, particularly if other platforms do not include similar capabilities or if the evaluation focuses on particular demographics who make up the bulk of the platform's users (e.g., young people, knowledge workers). For example, OtherTube piggybacked on YouTube via a desktop-only Chrome extension, which did not allow for understanding of the impact of its recommendation technique on mobile video browsing typical of young people [8].

In addition to these customization barriers imposing limitations on platforms and interactions, they also limit the kinds of social interactions which are even possible to evaluate through piggybacking. Because we can only examine existing examples of piggyback prototyped systems, it is less evident what research questions have been abandoned or not explored due to a lack of platform support.

6.2 Ethical tensions

Repurposing systems that people use for everyday communication to instead be used for research introduces ethical questions. For researchers, this surfaces the question of how to balance participant privacy with the need to collect research data. At a broader level, it also surfaces a question around whether leveraging existing platforms needlessly subjects participants to the negative aspects of those platforms.

6.2.1 Violating privacy expectations. To answer research questions, the deployment of piggyback prototypes in research settings often necessitates observation of how, what, and with whom participants communicate using the prototype. Depending on the construction of the social system being piggybacked, access constraints can make it challenging to observe participant experiences. Grevet & Gilbert highlight that the public nature of Twitter's feed proved helpful in understanding whether or how participants use the prototype to converse with others [37]. However, requiring participants to communicate on a public channel can also violate privacy. Suh et al. point out that

while public participant actions can be beneficial for research observation, there is also risk when data being shared is sensitive, requiring informing participants of the risk or providing privacy controls [96].

Piggybacking off of systems that do not make conversations publicly available requires researchers to take alternate approaches to understand how participants use the system. Understanding participants' experiences can necessitate invading participant's social circles or conversations. In some studies, researchers chose to rely on participants to share their private experiences. WhatFutures designated a "Future Guide" whose goal was to collect what participants shared to the group [56]. Other systems noted that a researcher joined the groups the systems created to observe and collect what participants were conversing about, such as private WhatsApp or Facebook groups [25, 57]. In all examples we noted, this conversational data collection was disclosed and highlighted to participants ahead of time. Nonetheless, this research practice may come in tension with people's expectations around how conversations on that platform may be used.

Invasive participant's private conversations may also influence what participants are willing to disclose. For example, researchers being in Facebook groups in the ARC method may have impacted participant's discussions on sensitive topics like HIV experiences [74]. Cho et al. similarly suggest that recording participant's KakaoTalk messages may have influenced the conversations they were willing to have with others [13].

Piggyback prototyping introduces a risk of unintended data disclosures, both to the research team and to the platforms being piggybacked on top of. The design of piggyback prototypes may result in collecting data from people who did not explicitly consent to being a part of the study. For example, piggyback prototypes can produce posts to a public or semi-public social feed and analyze responses from audience members, such as in CommitToSteps [78]. Although prototypes may clarify that the post was made as part of a research study and that engagement may be analyzed, Grevet & Gilbert note the difficulties of obtaining explicit consent in the context of piggybacking and suggest that waivers of documentation may be sufficient [37]. However, many people do not believe that their public social media activity should be used without express consent [30].

Although participants may be comfortable sharing content with the research team, piggybacking on existing social media systems presents a risk of disclosing participant information to the social media platform [53]. Munson & Consolvo expressed concern that their GoalPost app would disclose information about participants' exercise to Facebook, and therefore designed their piggybacking to include vague information in the Facebook posts themselves along with a link to a different site containing more details [77].

6.2.2 Reinforcing the negative aspects of social computing platforms. Widely-used social computing platforms have faced criticism for promoting a range of negative behaviors, including inciting harassment, lowering self-esteem, and spreading disinformation [46]. Requiring participants to engage with these platforms to use piggyback prototypes risks exposing them to this negative content during research studies. At a more meta-level, researchers building on and engaging with these platforms has the potential to provide endorsement of the platform's features and the creators' practices.

More directly, piggyback prototyping can potentially perpetuate exposure to negative content by promoting unethical practices. For example, researchers introducing or popularizing technical loopholes needed to conduct research work, such as publicizing a person's private activity, could make them more likely to be exploited by less benevolent parties. A social platform may further incorporate a participant's behavior in the piggyback prototype into later recommendations it provides (e.g., recommending advertisements influenced by their study activity, such as for HIV medication). Additionally, many people have opted out of commercial social platforms to avoid the

content associated with those platforms. Piggybacking could therefore introduce a sample bias or a barrier to participation in research, limiting the generalizability of findings.

The CSCW community is continuing to discuss whether we should aim to iteratively improve existing social technologies to address these exclusionary practices or instead aim to develop new social technologies which do not encode them [98]. While the community may agree on the overall goal (make social computing systems better for the people who use them), epistemological and philosophical perspectives may result in different perspectives on how to achieve that goal. Towards the vision of improving existing technologies, piggyback prototyping provides one potentially productive model for evaluating modifications to current social technologies which can improve on these negative experiences. However, it is important to acknowledge the limits of the method on reforming significant issues around these platforms.

7 CONSIDERING BENEFITS AND DRAWBACKS THROUGH THREE CASE STUDIES

To further surface how the benefits and drawbacks of piggyback prototyping influence researchers' design choices, system development, and evaluation, we discuss three case studies of piggyback prototyping where we are familiar with the design and evaluation process. These case studies provide examples of how the method enables researchers to address design challenges that are otherwise difficult to address, but also introduce limitations impacting design or evaluation.

7.1 SnapPI: Data-Driven Stickers for Ephemeral Social Media

Our first case study, SnapPI (Figure 1), supports creating stickers with personal data, such as physical activity and heart rate, for affixing to photos and videos shared on Snapchat [101]. Prior systems used piggyback prototyping to support sharing similar personal data to more archival social media (e.g., Facebook), finding that participants who used these systems were often concerned that the moments they collected with personal data were too trivial to share [25, 26, 77, 78]. SnapPI was motivated by research suggesting that ephemeral platforms might be an effective space for balancing the benefits of sharing everyday data with concerns that the data does not represent significant enough accomplishments to warrant sharing on archival social media [27]. Designing and evaluating SnapPI allowed for understanding when and how people would like to incorporate personal data into their everyday conversations with friends and family.

From a technical perspective, a key benefit of integrating into an existing platform enabled SnapPI to leverage many of the customization and sharing options that Snapchat provides. For example, participants playfully connected their data stickers with the photos that they shared. For example, a participant manipulated a “fork” sticker indicating how many calories they were eating to appear as though it was on top of the plate they had taken a photo of. These features enabled participants to transfer their communication style with how they use Snapchat's existing stickers to how they used SnapPI's data stickers.

Intellectually, the lack of friction when sharing with friends and family members deepened the researcher's ability to understand how people would like to use personal data to communicate. Participants appreciated how SnapPI stickers could naturally be integrated into the conversations they were already having on the platform. In particular, the researchers noted that the data sharing SnapPI supported led participants to have new conversations with their friends and family around health and wellbeing, as they were able to incorporate relevant stickers into the everyday conversations that they had on the platform.

In developing SnapPI, technical barriers influenced the design of the tool. Although multiple social media sites support ephemeral sharing and augmenting photos and videos with stickers (e.g., Instagram and Facebook Stories), Snapchat was the only widely-used platform with an API that supported importing external stickers for use within the platform. Consequently, participant

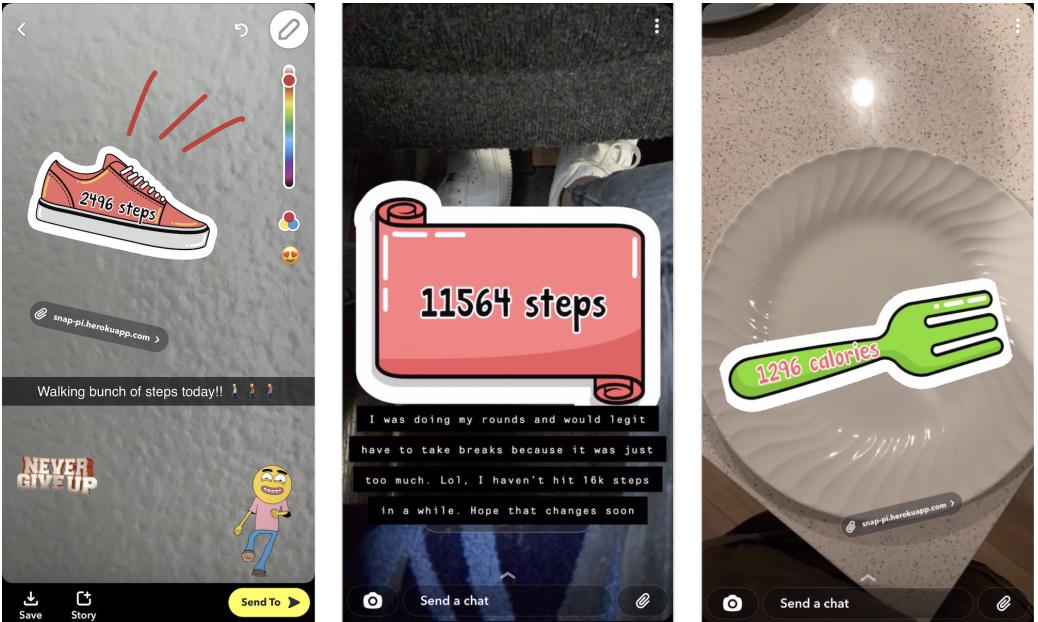


Fig. 1. SnapPI supported creating data-driven stickers for sharing on Snapchat. Piggybacking off of Snapchat allowed the researchers to leverage the platform’s existing features, like annotation with other stickers and text, and supported understanding of how people wished to integrate stickers into everyday conversations.

recruitment became challenging, as eligible participants needed to both be active enough on Snapchat and have enough experience collecting personal data to minimize novelty effects of sharing data ephemerally. Beyond limiting eligible participants, the nature of social media ecologies [109] meant that data was absent or difficult to integrate into conversations participants had with their Snapchat contacts off the platform.

Aligning with the expectation of polish, research suggests the need for tools that share personal data to support customization of visuals, text, and numbers to avoid repetition in sharing [26, 64, 77]. To support this goal, the authors implemented a relatively large number of stickers (39) and customization options (e.g., sticker color, unit of measurement like “steps” and “miles”). Other customization options, such as font and text color, required development work that fell out of scope (e.g., wrapping text within a sticker, ensuring legibility). Participants appreciated the sticker and customization options and used them widely. However, multiple participants still feared that the content they shared would eventually become boring if they continued using SnapPI beyond the study duration, and suggested that additional stickers and customization options would help.

Accessing participants’ use of SnapPI required violating both personal and platform privacy expectations. Because stickers were exported from SnapPI to Snapchat, SnapPI’s telemetry was unable to record how participants embedded the sticker in their Snap, whether participants sent a Snap with the sticker at all, who they sent it to, or how audience members responded. The authors, therefore, asked participants to send their Snaps to an account associated with the research team and took screenshots for later analysis. Although this enabled participants to filter out Snaps containing photos or information they did not want to disclose to the researchers, some participants frequently forgot to share their Snaps with the research team. Additionally, the ephemerality of Snapchat creates expectations that posts will disappear [104]. The researchers taking screenshots

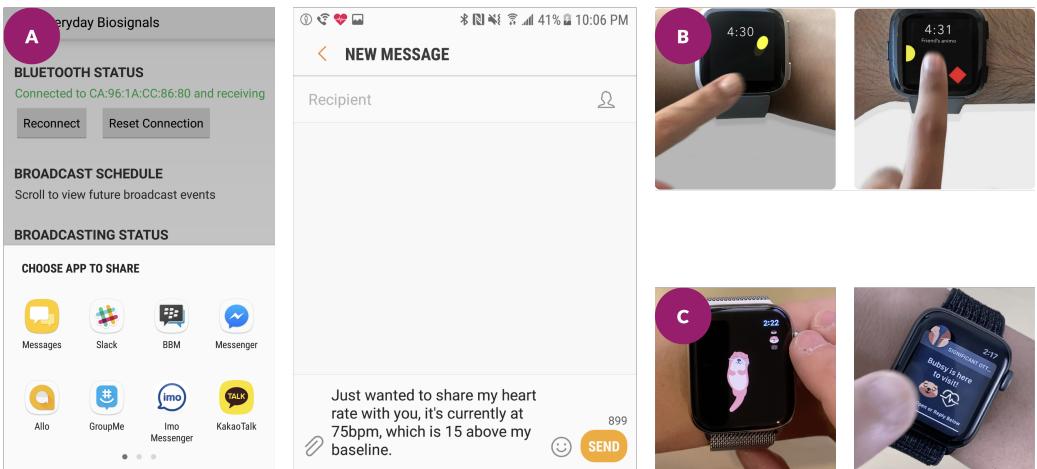


Fig. 2. The (A) Everyday Biosignals system supported people in sharing their heart rate via messaging apps, enabling understanding of biosignal use in everyday conversation within chosen communication channels. Conversely, (B) Animo and (C) Significant Otter implemented standalone systems for communicating biosignals, giving greater design flexibility by removing the format constraints of existing channels.

for later analysis violated this expectation, and may have influenced how participants perceived or used the platform.

7.2 Expressive Biosignals

Our second case study focuses on prototypes that integrate *expressive biosignals*, or the display of sensed physiological data in social interactions [63]. Motivated by the limited nonverbal cues available in remote communication, as well as the increasing ubiquity of consumer-grade wearable sensors, this line of research introduced biosignals as a new social cue that technology can afford. Liu et al. investigated several research questions around understanding people’s decisions behind sharing such data and their social impact in dyadic communication. For the purposes of this paper, we describe and compare three expressive biosignals systems they built that vary in their degree of piggyback prototyping: an “Everyday Biosignals” text messaging system [64], the Animo Fitbit Versa watchface [65], and the Significant Otter iPhone/Apple Watch app [66] (Figure 2).

7.2.1 Using piggyback prototyping. Everyday Biosignals is an Android app that prompts users to text their heart rate to their contacts using a messaging app of their choice (e.g., built-in SMS, WhatsApp, Snapchat). Liu et al. built and deployed this system to explore people’s sharing motivations and behaviors around biosignals like heart rate [64]. The app connects to the Mio Alpha 2 watch via Bluetooth LE in order to continuously read heart rate, which is used to send notifications (i.e., when heart rate is high or low compared to the user’s baseline) asking the user if they would like to share their current heart rate with a contact. If the user selects “yes,” they can then select a messaging app of their choice, which the system opens and pre-populates with editable text containing their heart rate in beats per minute (bpm). As such, the system piggybacks off of these existing messaging apps to enable users to message their heart rate to others. The authors found that with this system, people shared their heart rate in order to express their emotions and give status updates, but the heart rate bpm was not engaging by itself and required people to apply their own meanings to it.

The authors' decision to piggyback off of existing messaging apps was due to the ability of users "*to share with their contacts in a naturalistic fashion using their typical communication channels.*" Since people use different communication apps for different people or contexts [83], leveraging their existing apps reduced the need for participants to learn and incorporate a new channel into their communication, and allowed them to more naturally express themselves on platforms they typically used with any of their contacts. This enabled high ecological validity, where participants considered heart rate sharing in light of their real-world communication practices. Subsequently, like SnapPI, users embedded their heart rate into conversations they already have with others. The researchers could thus address one of their key questions: "*how can people meaningfully express their heart rate to others?*"

Because Everyday Biosignals piggybacked off of other communication apps, Liu et al. asked participants to send the researchers screenshots of the conversations they had in their communication apps when they shared their heart rate. The researchers could therefore observe how people spoke about their heart rate with others, including how they edited the pre-populated heart rate text, provided context or explanation for the text, or conversed about the data. At the same time, using screenshots limited the ability to instrument how participants used the various chat apps to converse around their biosignals. For instance, the screenshots were sometimes insufficient to understand the context of the conversations that participants had around the shared biosignals.

7.2.2 Building a standalone system. In contrast to Everyday Biosignals, Liu et al. also created two new systems which did not piggyback off of existing social computing systems: Animo [65] and Significant Otter [66]. Both are smartwatch apps that enable users to share their heart rate with each other in the form of an animated avatar. The authors found that sharing heart rate in these ways enabled people to feel connected with close others in a lightweight way throughout their daily lives. Unlike Everyday Biosignals, these systems are new social computing apps that the authors built and deployed to understand the effects of biosignals sharing.

Piggyback prototyping with Everyday Biosignals had the benefits of exploring biosignals as integrated into natural texting behaviors. In contrast, Animo and Significant Otter brought the opportunity to explore biosignals sharing as its own communication channel. In particular, developing standalone applications enabled greater design freedom, without being tied to the format of existing messaging apps (i.e., text). Since Everyday Biosignals' raw text heart rate in bpm was not very engaging, the authors designed more playful representations for biosignals – specifically, shapes or otters that animated according to high or low heart rate. Additionally, exploring these representations outside of back-and-forth text conversations allowed biosignals to act as lightweight cues on their own, which helped participants keep up-to-date with each other and start new conversations at later times. The authors also explored the potential for sharing biosignals on a new form factor: smartwatches. This form factor enhanced ease-of-use, as participants could quickly glance at and tap their avatar on their watchface to send it, as well as feelings of presence, where its location on the wrist gave a sense of being physically "touched." Leveraging this form factor would not have been possible through piggybacking, as there were very few existing social watch apps at the time, and none that allowed for customizing the shared watchface via piggybacking. Finally, developing a new app created a unique channel between participants and their partner, which enabled a more intimate dyadic communication experience.

At the same time, qualitative results from these works reveal some drawbacks to building a standalone app. A few participants wished that the app was more integrated with their existing messaging apps, such that they could use their established forms of communication with their partner or other people outside of the study. For example, they described wanting to paste their avatar into their texts rather than only view and send it on their watch. Some participants emulated

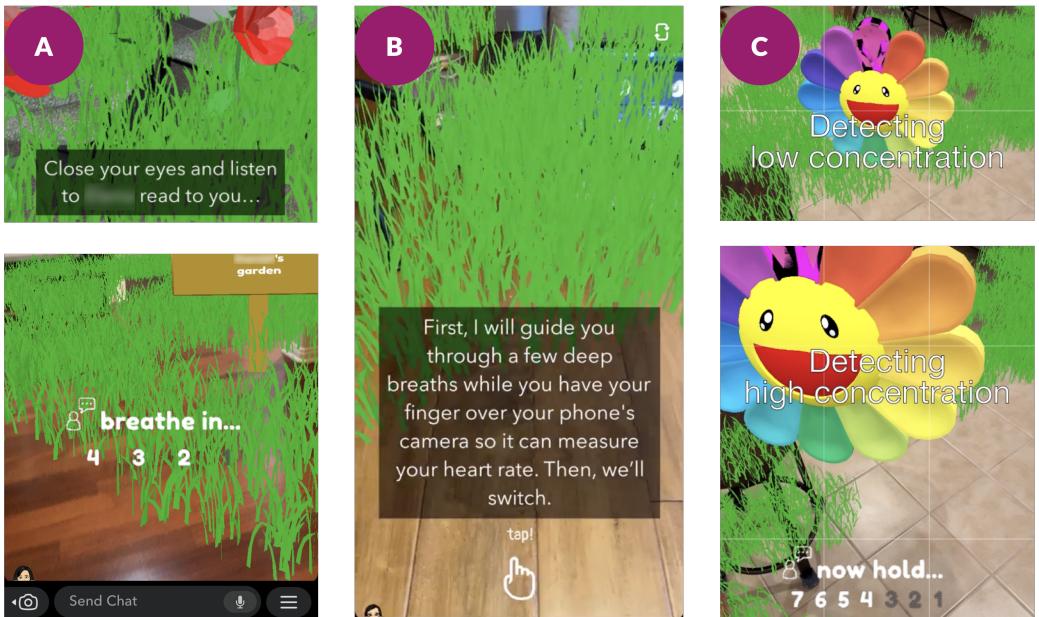


Fig. 3. Mindful Garden piggybacks off of Snapchat, leveraging the platform’s Augmented Reality framework to (A) support a co-located mindfulness session where a virtual garden is populated by flowers correlating with biosignals detected from the camera. (B) In the first version, heart rate is detected by having the person being guided place their finger over the phone’s camera. (C) In the second version, more sophisticated biosignals are detected from an external device, requiring data to be transferred to Snapchat through a covert channel.

this by sending their avatar and immediately sending a follow-up text to explain that avatar. This is similar to Everyday Biosignals, where people used the conversation to provide context for ambiguous biosignals. Thus, while building standalone systems had many benefits that outweighed these drawbacks, Liu et al. still highlight integration with existing social computing systems as a major design implication.

7.3 Mindful Garden: Co-located Social Interaction in Augmented Reality

Mindful Garden is an ongoing project exploring how augmented reality on mobile phones and headsets, in conjunction with biosignal data, can help people who are co-located in a physical space connect and share with one another [67]. Co-located mindfulness meditation has the potential to more deeply connect the people meditating, giving space for them to jointly influence their environment. Biosignals can offer feedback on how mindful a person is being, and can help make people feel closer to one another. In Mindful Garden, pairs of people guide each other through a breathing exercise, lasting about 1 minute per person. In augmented reality, both participants are presented a garden containing flowers which grow correlated with each participant’s biosignals (Figure 3). Mindful Garden is built as a Lens on Snapchat with Lens Studio [95] to leverage the platform’s support for co-located augmented reality and minimize the setup burden of installing a new app.

Two versions of Mindful Garden are currently being developed. In the first version, the number of flowers are correlated with the heart rate of the person receiving the meditation guidance. To infer heart rate, person receiving meditation guidance places their finger over their phone’s camera for the system to detect differences in average pixel color. This allows the system to operate entirely on

a mobile device within the Snapchat app. In the second version, participants' flowers are correlated with heart rate and brain wave data from a Muse 2 headset, with indicators like the flower's growth rate, color, and number of petals correlated with specific brain waves.

The benefits of piggybacking Mindful Garden off of Snapchat have been primarily technical. Preliminary findings suggest that piggybacking was helpful for easing the burden of setup, as the system was distributed as an add-on to the social platform (via a Snap Code) rather than requiring separate installation. It also allowed for building off of the well-tested and complex development environment within Snapchat for augmented reality, reducing the technical burden of making the Lenses.

The challenges of piggyback prototyping in Mindful Garden have also been primarily technical. As of the time of writing, Lens Studio does not support HTTP requests for accessing external data, beyond information Snap already collects about a person (e.g., name, birthday) or their environment (e.g., altitude, temperature, time, weather) [93]. To incorporate biosignals, both versions of Mindful Garden therefore make use of input signals supported by Lens Studio. The first version leverages raw pixel data from the camera to directly measure heart rate. For the second version, the researchers tested a few techniques to transfer biosignal data from the Muse 2 headset to the lens. One technique involved displaying a digital tag (e.g., a QR code, a recognizable object) that encoded the data on a device such as a tablet in the co-located environment. While this technique was technically feasible, it required participants to closely point the camera at the tag to read it. This interfered with the ability to render a garden in the environment, as most of the visible environment was the tag, and also would distract the participant from the mindful activity. The technique ultimately used was to have the device in the environment play a tone at an inaudible frequency which encodes the biosignal values from the Muse device, which the Lens can interpret.

These technical challenges have resulted in less desirable user experiences. In the first version, leveraging the camera as an input source resulted in relatively low accuracy of heart rate measurement. Consequently, people using the app tend to interpret the flower growth as random, rather than being connected to their heart rate. While heart rate measurement has been more accurate in the second version, the indirect transfer of the biosignal has only allowed a few discrete values to be transmitted. Because audio detection in the Lens was primarily intended to capture audible noise, a relatively narrow band of audio frequency was both inaudible and detectable by the Lens. This led to limitations around the fidelity of data that could be transmitted, only allowing biosignals such as level of concentration to be regarded as either "low" or "high", rather than the continuous input that the Muse 2 headset provides. This discretization has constrained some ideas around encoding the biosignals into the garden design.

8 DISCUSSION

Through reviewing the goals, benefits, and tensions of piggyback prototyping social communication experiences, we have extended Grevet & Gilbert's focus on assessing critical mass of interest [37] to demonstrate how the method can promote other research goals including the ordering and filtering of social content or formation of affinity groups. We articulate how the method can increase the ecological validity of research studies by leveraging familiarity, enabling more interesting and realistic social relationships, and structuring participant data collection. We also describe challenges to deploying the method, including technical limitations, expectations and disclosure concerns. Here, we briefly outline points of consideration when deciding whether to piggyback off of an existing social system. We then advocate for wider use of the method, and finally, describe a few potential research agendas to support piggyback prototyping.

8.1 Considerations When Deciding Whether to Piggyback

We suggest three considerations when deciding whether piggybacking off of an existing social system is more effective for answering a research question than developing a social system from scratch.

8.1.1 Is my research question better answered by leveraging an existing platform or starting from scratch? A first consideration is whether piggyback prototyping is beneficial in answering the intended research question. We believe that piggyback prototyping has significant advantages over starting from scratch for answering research questions around how people perceive a new technique or capability. Specifically, it can allow for a deeper understanding of the everyday considerations, constraints, and circumstances people will inevitably face if the technique being prototyped were present in social computing systems people use regularly. However, piggybacking can constrain the design space of social interactions to existing tools' paradigms, limiting piggybacking if the research question proposes a radically different approach to communication. For example, it might prove difficult or limiting to design for sharing some types of content through piggybacking given the capabilities of existing systems, like prototyping social experiences in virtual reality.

A related question to consider is whether people are already communicating on an existing platform, or are interested in doing so. The main benefit of piggyback prototyping is minimizing the gap between people's everyday use and research use, suggesting value in selecting platforms where the relevant stakeholders have already congregated or are interested in congregating. Further, considering whether people's use of the platform aligns with the research goal can help reduce friction to use, as people have preferred platforms for communicating with specific people or about specific topics [83]. For example, a researcher interested in testing new mechanisms for sharing messages of affection may benefit from piggybacking off of a direct communication tool like SMS rather than a productivity communication tool like Slack, while the reverse might be preferable for testing a creative ideation communication workflow.

A potentially valuable practice is to survey or interview the target audience about platform interest and appropriateness before designing, which can influence the choice of platform to piggyback off of [56, 71]. For conducting studies, researchers should also consider whether they will be able to recruit participants who are interested in and able to communicate on the platform and meet other inclusion criteria. Piggybacking off of multiple platforms may allow for greater flexibility, but comes with an additional development burden.

8.1.2 Do any platforms allow for and simplify the design approach I am considering? With benefits assumed, a key practical question is whether it is even possible to extend an existing system as desired. In particular, we recommend establishing whether a given customization is fundamentally possible before investing design and development resources into implementing the method. Many systems leverage hacks, loopholes, or other workarounds to accomplish the goal, such as transmitting biosignals over audio in Mindful Garden. In other systems, a less-optimal user experience was supported because the intended customization was not possible [106]. In all of these cases, an important consideration is to what extent the workaround will be noticeable by a potential participant or user, or to what extent it will negatively impact the experience, as limited or impacted participant use can impact researcher's ability to answer their research questions.

A range of technical options can be used for piggyback prototyping. Researchers prototyping on top of systems on the web might consider browser extensions. Browser extensions are well-suited for piggybacking as they are not dependent on the underlying system having extensible features or APIs and only impact the person who has the extension installed, allowing for asymmetric experiences. In extreme circumstances, researchers could leverage pixel-based techniques to enable

advanced customizations [22]. In these cases, it is worth considering whether piggybacking reduces the development burden beyond building a social computing system from scratch. Alternatively, for contributions where empirical evaluations are not needed, it may be sufficient to provide a technical sketch of how such an extension would work on a particular platform [107].

Prior uses of piggyback prototyping also emphasize that many research questions can be answered solely through repurposing a platform’s built-in mechanics, such as creating a researcher-organized Facebook Group [73, 74] or structuring a Google Doc to support the desired communication task [84]. If existing mechanics are sufficient for achieving the intended research goal, it can help reduce development burden and increase participant familiarity.

Beyond viability, it is important to consider whether piggybacking is worth the risks inherited from building on an existing platform. A piggyback prototype study can become completely upended if a commercial API is adjusted to revoke a necessary feature, if a website is adjusted in a way that breaks a browser extension, or if a system is taken down entirely. While deployments of standalone social computing systems in the field are frequently disrupted by researcher-created bugs [92], the researcher is not as beholden to the whims of the creators of the system being piggybacked on, and has enough control to address issues mid-deployment. On the flip side, piggybacking can inherit some reliability from well-tested commercial systems, such as minimizing concerns about whether messages will be delivered from one person to another.

As a generalization, our inclination is that features of a system are likely safe to piggyback on if non-researchers rely on the techniques. For example, SnapPI leveraged Snap’s Creative Kit, which advertisers use to create custom stickers and lenses for their websites and products [94]. This instills trust that, in theory, some warning would be issued prior to sunsetting the API. For platforms that may frequently change or A/B test their interfaces or features with minimal or no warning (e.g., websites used by browser extensions), we believe that the relatively fast development cycles of most piggyback prototypes (e.g., a year or two) and relatively short durations of deployment studies (e.g., weeks or a month) can lower risks. However, we expect that the risks are likely less worth inheriting in social computing systems which aim to exist long-term and undergo frequent iteration. For example, reflecting on a decade of Python Tutor [39, 40, 42], Guo identified that the reliance on extremely minimal software dependencies enabled greater reliability of the social system [41].

8.1.3 Does piggybacking in this domain introduce undue privacy or ethical concerns? As discussed, the nature of piggybacking on social platforms provides access to the conversations that participants are having, which in many cases is part of the research data. If those conversations are sensitive, it may not be desirable to disclose them. Additionally, as pointed out by prior work, people’s public or semi-public behaviors may become part of the research data [37, 78]. We recommend that researchers creating piggyback prototypes clarify what information about engagement is being collected, both with participants and with anyone who may engage with participant-generated content. In cases where the privacy or ethical risks of piggybacking exceed the benefit of leveraging an existing platform, a standalone system may need to be developed or the research question may need to be refocused.

8.2 Advocating for piggybacking and piggybackable platforms

In outlining a set of benefits that piggyback prototyping provides for answering a range of CSCW systems questions, we argue that the community could benefit from broader use of the method relative to designing single-purpose research systems. We also suggest that making more social systems piggybackable can enable further research and experimentation, and deepen our knowledge

of how to support people's social communication. We advocate for supporting extensibility of both commercial social communication systems and research prototypes.

For commercial systems, supporting extensibility can allow the broader community to envision, design, evaluate, and advocate for communication techniques that are not on the critical path of the platform. For example, the Boomerang extension for schedule-sending emails inspired a technique for improving how people use Gmail to communicate with one another, which was eventually integrated into the platform [97]. Similarly, TweetDeck's dashboard was eventually acquired by Twitter, and some features were subsequently integrated into the native app [24].

For research systems, the benefits roughly align with the benefits of supporting extensibility or comparison of other research systems. For example, if a research app that implements a new kind of communication were released as open-source or with extensible APIs, it could support more direct comparisons, such as allowing A/B testing between the previous and new technique.

In spite of the benefits, it is important to acknowledge the risks and downsides of supporting this sort of extensibility. For research systems, requiring extensibility further increases the bar of contributing to a social system. The support and maintenance of such a system, while admirable and valuable, may not align well with the expectations of the early-career academics who frequently build them [41]. For commercial systems, supporting extensibility inevitably creates an opportunity for others to misuse or abuse the platform, with Cambridge Analytica being one prominent example [16].

8.2.1 Acknowledging the difficulty of piggybacking social systems. Although we see the benefit in piggybacking as a method for evaluating new concepts in social systems, we also wish to point out the difficulties of executing it. While researchers may be able to set up some expectation that issues will arise in field deployments of prototypes [92], participants using piggyback prototypes are occasionally less forgiving of issues with the prototype. These issues also have the potential to impact participants' social relationships as part of everyday use. They may worry how their friends and family might interpret their use- or non-use when things go wrong, such as when messages are sent multiple times or not sent at all. Rather than simply inconveniencing the participant and impacting the researcher's ability to collect useful perspectives, piggyback prototypes embed the researcher in the participants' social system with all of its complexities and expectations. Because piggybacked systems often leverage workarounds or hacks to implement their novel social communication techniques, these technical and usability issues may be more frequent than in other kinds of prototypes. Addressing these challenges can require additional testing or development beyond the high expectations of deployment studies, or may not be addressable at all.

It may be possible to partially couch or focus participant's feedback through more targeted approaches than strictly measuring or experimentally comparing use with or without the prototype. Qualitative methods like interviews or surveys can help give participants space to articulate what felt unpolished or could be streamlined alongside feedback on the core idea that the prototype aimed to evaluate.

Relatedly, Bernstein et al. note difficulties articulating the novelty of a social computing systems [4]. We echo Bernstein et al.'s suggestion that piggyback prototypes, like other social computing systems, can be evaluated on their *social contributions*, considering whether the prototype enables some new social interaction. Past that point, as Bernstein et al. suggest, any voluntary use by participants can be viewed as a success, and should be viewed by the research community as such.

8.3 New Research Agendas Suggested By Piggyback Prototyping

The widespread interest and use of piggyback prototyping points to valuable new research agendas and directions for future CSCW systems researchers, specifically in the need for richer toolkits. One

potentially valuable direction for future research is in tools to support researchers and developers looking to extend existing platforms. These tools would primarily aim to lower the threshold to developing piggyback prototypes [80], providing APIs which implement common features needed for piggybacking or contribute methods for piggybacking without programming. While prior piggyback prototypes occasionally leveraged existing third-party toolkits, such as existing chatbot frameworks [71, 106], we found that most implemented their prototype either from scratch (e.g., via manipulating a web page’s DOM) or with the help of the platform’s official APIs. Other spaces have demonstrated the utility of researcher-created toolkits for common problems, such as researchers piggybacking off of TurKit [62] to prototype crowdsourcing workflows for Amazon Mechanical Turk [6, 9, 12].

Because many piggyback prototypes aim to support creation of shareable content, research contributing tools for authoring content on social systems could support future piggyback prototypes. For example, many researchers have suggested that their prototype would benefit from more customizable shareable content [26, 38, 77]. Toolkits could help both researchers and participants by providing common customization options such as editable text, image layout, colors, emoji, and fonts, and support exporting them to social platforms. However, understanding the design requirements for such a tool would require further research. Additionally, tools could aggregate authoring features from different commercial APIs to allow studies to recruit participants who have different platform preferences. In doing so, tool designers could explore ways of reducing the burden to developers answering whether platforms allow for the planned customization, such as considering how to surface the content authoring limitations of different platforms.

When the platforms being piggybacked on do not provide APIs for accessing or manipulating shared content, piggyback prototypes often develop pipelines for accessing and editing content such as browser extensions. We observed some patterns implemented across a few different prototypes, such as accessing the video of a livestream [69, 105]. By understanding the needs of developers looking to extend these systems and developing tools for supporting them, researchers could contribute paradigms for extending existing platforms.

Tools can also help researchers configure existing social communication systems around affinity groups, supporting paradigms like the ARC method [73, 74, 86] and WhatFutures [56, 57]. Tools could promote some of the effective design choices made in prior instantiations of the methods, such as bringing up how the names of groups can impact privacy [74]. Tools could also support automating tasks that APIs allow for, but prior researchers have manually undertaken to avoid development burden. For example, tools could provide mechanisms for scheduling daily activities in the ARC method rather than requiring researchers to post them [73], and tools could automatically collect what participants communicate to one another in the group’s social feed.

Beyond toolkits, it is also worth considering whether methods like Wizard of Oz could be used to avoid technical barriers to piggyback prototyping while still evaluating the core research questions [35]. For example, a novel approach for summarizing content might be able to avoid a limitation around programmatic access to existing posts in a social computing system by a researcher manually copying posts over, within reason. Similarly, studying a new kind of social content could avoid limitations around programmatic authoring of content, by giving participants text, images, etc. which can be copied directly into the platform. Designing such Wizard of Oz methods, and demonstrating scenarios where they could support rapid ideation and evaluation of design ideas for social computing systems, is a promising direction for complementing piggyback prototypes.

9 CONCLUSION

In reconsidering and expanding the definition of piggyback prototyping, we demonstrate that in addition to assessing critical mass of interest on a topic in a social network, the method is helpful for answering research questions on the design of social communication features. We describe how the method provides ecological benefits over developing standalone prototypes, but also introduces ethical challenges around who can access participant conversations during studies. We encourage social computing systems researchers to consider whether the method is better-suited to answering their research questions than developing a standalone system, and suggest the research agenda of developing tools to reduce the burden of applying the method.

ACKNOWLEDGMENTS

We thank our coauthors of SnapPI, Expressive Biosignals, and Mindful Garden and participants at the Social Computing Systems Summer Camp for initial discussions on this topic. This work was supported in part by the National Science Foundation under award IIS-1850389, the UCI Council on Research, Computing, and Libraries (CORCL), and the Snap Creative Challenge on the Future of Co-located Social Augmented Reality.

REFERENCES

- [1] Elena Agapie, Daniel Avrahami, and Jennifer Marlow. 2016. Staying the Course: System-Driven Lapse Management for Supporting Behavior Change. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2016)*. 1072–1083. <https://doi.org/10.1145/2858036.2858142>
- [2] Saleema Amershi, James Fogarty, and Daniel Weld. 2012. Regroup: Interactive Machine Learning for On-Demand Group Creation in Social Networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2012)*. 21–30. <https://doi.org/10.1145/2207676.2207680>
- [3] Paul André, Michael Bernstein, Mira Dontcheva, Elizabeth Gerber, Aniket Kittur, and Rob Miller. 2012. CrowdCamp: Rapidly Iterating Ideas Related to Collective Intelligence & Crowdsourcing. In *Extended Abstracts on Human Factors in Computing Systems (CHI 2012 EA)*. 2687–2690. <https://doi.org/10.1145/2212776.2212696>
- [4] Michael S. Bernstein, Mark S. Ackerman, Ed H. Chi, and Robert C. Miller. 2011. The Trouble with Social Computing Systems Research. In *Extended Abstracts on Human Factors in Computing Systems (CHI 2011 EA)*. 389–398. <https://doi.org/10.1145/1979742.1979618>
- [5] Michael S. Bernstein, Irene Greif, Wendy Mackay, Hiroshi Ishii, Jonathan Grudin, Karrie Karahalios, Meredith Ringel Morris, Aniket Kittur, Jaime Teevan, Amy X. Zhang, and Niloufar Salehi. 2020. UIST+ CSCW: A Celebration of Systems Research in Collaborative and Social Computing. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2020)*. 1–3. <https://doi.org/10.1145/3379337.3422374>
- [6] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2010. Soylent: a Word Processor with a Crowd Inside. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2010)*. 313–322. <https://doi.org/10.1145/1866029.1866078>
- [7] Michael S. Bernstein, Adam Marcus, David R. Karger, and Robert C. Miller. 2010. Enhancing Directed Content Sharing on the Web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2010)*. 971–980. <https://doi.org/10.1145/1753326.1753470>
- [8] Md Momen Bhuiyan, Carlos Augusto Bautista Isaza, Tanushree Mitra, and Sang Won Lee. 2022. OtherTube: Facilitating Content Discovery and Reflection by Exchanging YouTube Recommendations with Strangers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2022)*. 1–17. <https://doi.org/10.1145/3491102.3502028>
- [9] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and Tom Yeh. 2010. Vizwiz: Nearly Real-Time Answers to Visual Questions. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2010)*. 333–342. <https://doi.org/10.1145/1866029.1866080>
- [10] Virginia Braun and Victoria Clarke. 2006. Using Thematic Analysis in Psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101. <https://doi.org/10.1191/1478088706qp063oa>
- [11] Eshwar Chandrasekharan, Chaitrali Gandhi, Matthew Wortley Mustelier, and Eric Gilbert. 2019. Crossmod: A Cross-Community Learning-Based System to Assist Reddit Moderators. *Proceedings of the ACM on human-computer interaction* 3, CSCW (2019), 1–30. <https://doi.org/10.1145/3359276>

- [12] Lydia B. Chilton, Greg Little, Darren Edge, Daniel S Weld, and James A. Landay. 2013. Cascade: Crowdsourcing Taxonomy Creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2014)*. 1999–2008. <https://doi.org/10.1145/2470654.2466265>
- [13] Hyunsung Cho, Jinyoung Oh, Juho Kim, and Sung-Ju Lee. 2020. I Share, You Care: Private Status Sharing and Sender-Controlled Notifications in Mobile Instant Messaging. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–25. <https://doi.org/10.1145/3392839>
- [14] John Joon Young Chung, Hijung Valentina Shin, Haijun Xia, Li-yi Wei, and Rubaiat Habib Kazi. 2021. Beyond Show of Hands: Engaging Viewers via Expressive and Scalable Visual Communication in Live Streaming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2021)*. 1–14. <https://doi.org/10.1145/3411764.3445419>
- [15] Giovanni Luca Ciampaglia and Dario Taraborelli. 2015. MoodBar: Increasing New User Retention in Wikipedia through Lightweight Socialization. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW 2015)*. 734–742. <https://doi.org/10.1145/2675133.2675181>
- [16] Nicholas Confessore. 2018. *Cambridge Analytica and Facebook: The Scandal and the Fallout So Far*. <https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html>
- [17] Justin Cranshaw, Emad Elwany, Todd Newman, Rafal Kocielnik, Bowen Yu, Sandeep Soni, Jaime Teevan, and Andrés Monroy-Hernández. 2017. Calendar.help: Designing a Workflow-Based Scheduling Agent with Humans in the Loop. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2017)*. 2382–2393. <https://doi.org/10.1145/3025453.3025780>
- [18] Franco Curmi, Maria Angela Ferrario, Jen Southern, and Jon Whittle. 2013. HeartLink: Open Broadcast of Live Biometric Data to Social Networks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1749–1758. <https://doi.org/10.1145/2470654.2466231>
- [19] Ella Dagan, Ana María Cárdenas Gasca, Ava Robinson, Anwar Noriega, Yu Jiang Tham, Rajan Vaish, and Andrés Monroy-Hernández. 2022. Project IRL: Playful Co-Located Interactions with Mobile Augmented Reality. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1 (2022), 1–27. <https://doi.org/10.1145/3512909>
- [20] Sayamindu Dasgupta and Benjamin Mako Hill. 2018. How “Wide Walls” can Increase Engagement: Evidence from a Natural Experiment in Scratch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2018)*. 1–11. <https://doi.org/10.1145/3173574.3173935>
- [21] Kushal Dave, Martin Wattenberg, and Michael Muller. 2004. Flash Forums and ForumReader: Navigating a New Kind of Large-Scale Online Discussion. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2004)*. 232–241. <https://doi.org/10.1145/1031607.1031644>
- [22] Morgan Dixon and James Fogarty. 2010. Prefab: Implementing Advanced Behaviors using Pixel-Based Reverse Engineering of Interface Structure. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2010)*. 1525–1534. <https://doi.org/10.1145/1753326.1753554>
- [23] Steven P. Dow, Kate Heddleston, and Scott R. Klemmer. 2009. The Efficacy of Prototyping under Time Constraints. In *Proceedings of the ACM conference on Creativity and cognition (C&C 2009)*. 165–174. <https://doi.org/10.1145/1640233.1640260>
- [24] Rip Emerson. 2011. Twitter Buys TweetDeck for \$40 Million. <https://techcrunch.com/2011/05/23/twitter-buys-tweetdeck-for-40-million/>
- [25] Daniel A. Epstein, Felicia Cordeiro, James Fogarty, Gary Hsieh, and Sean A. Munson. 2016. Crumbs: Lightweight Daily Food Challenges to Promote Engagement and Mindfulness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2016)*. 5632–5644. <https://doi.org/10.1145/2858036.2858044>
- [26] Daniel A. Epstein, Mira Dontcheva, James Fogarty, and Sean A. Munson. 2020. Yarn: Adding Meaning to Shared Personal Data through Structured Storytelling. In *Graphics Interface*. <https://doi.org/10.20380/GI2020.18>
- [27] Daniel A. Epstein, Siyun Ji, Danny Beltran, Griffin D’Haenens, Zhaomin Li, and Tan Zhou. 2020. Exploring Design Principles for Sharing of Personal Informatics Data on Ephemeral Social Media. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 1–24. <https://doi.org/10.1145/3415166>
- [28] Motahhare Eslami, Karrie Karahalios, Christian Sandvig, Kristen Vaccaro, Aimee Rickman, Kevin Hamilton, and Alex Kirlik. 2016. First I “like” it, then I hide it: Folk Theories of Social Feeds. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2016)*. 2371–2382. <https://doi.org/10.1145/2858036.2858494>
- [29] Motahhare Eslami, Aimee Rickman, Kristen Vaccaro, Amirhossein Aleyasen, Andy Vuong, Karrie Karahalios, Kevin Hamilton, and Christian Sandvig. 2015. “I always assumed that I wasn’t really that close to [her]” Reasoning about Invisible Algorithms in News Feeds. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2015)*. 153–162. <https://doi.org/10.1145/2702123.2702556>
- [30] Casey Fiesler and Nicholas Proferes. 2018. “Participant” perceptions of Twitter Research Ethics. *Social Media+ Society* 4, 1 (2018), 2056305118763366. <https://doi.org/10.1177%2F2056305118763366>
- [31] Denae Ford, Kristina Lustig, Jeremy Banks, and Chris Parnin. 2018. “We Don’t Do That Here” How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities. In *Proceedings of the SIGCHI Conference on*

- Human Factors in Computing Systems (CHI 2018)*. 1–12. <https://doi.org/10.1145/3173574.3174182>
- [32] Siwei Fu, Jian Zhao, Hao Fei Cheng, Haiyi Zhu, and Jennifer Marlow. 2018. T-Cal: Understanding Team Conversational Data with Calendar-Based Visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2018)*. 1–13. <https://doi.org/10.1145/3173574.3174074>
- [33] R. Stuart Geiger. 2014. Bots, Bespoke, Code and the Materiality of Software Platforms. *Information, Communication & Society* 17, 3 (2014), 342–356. <https://doi.org/10.1080/1369118X.2013.873069>
- [34] Eric Gilbert. 2012. Predicting Tie Strength in a New Medium. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2012)*. 1047–1056. <https://doi.org/10.1145/2145204.2145360>
- [35] John D. Gould and Clayton Lewis. 1985. Designing for Usability: Key Principles and What Designers Think. *Commun. ACM* 28, 3 (1985), 300–311. <https://doi.org/10.1145/3166.3170>
- [36] Saul Greenberg and Bill Buxton. 2008. Usability Evaluation Considered Harmful (Some of the Time). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2008)*. 111–120. <https://doi.org/10.1145/1357054.1357074>
- [37] Catherine Grevet and Eric Gilbert. 2015. Piggyback prototyping: Using Existing, Large-Scale Social Computing Systems to Prototype New Ones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2015)*. 4047–4056. <https://doi.org/10.1145/2702123.2702395>
- [38] Carla F. Griggio, Arissa J. Sato, Wendy E. Mackay, and Koji Yatani. 2021. Mediating Intimacy with DearBoard: a Co-Customizable Keyboard for Everyday Messaging. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2021)*. <https://doi.org/10.1145/3411764.3445757>
- [39] Philip J. Guo. 2013. Online Python Tutor: Embeddable Web-Based Program Visualization for CS Education. In *Proceeding of the ACM Technical Symposium on Computer Science Education (SIGCSE 2013)*. 579–584. <https://doi.org/10.1145/2445196.2445368>
- [40] Philip J. Guo. 2015. Codeopticon: Real-Time, One-to-Many Human Tutoring for Computer Programming. In *Proceedings of the ACM Symposium on User Interface Software & Technology (UIST 2015)*. 599–608. <https://doi.org/10.1145/2807442.2807469>
- [41] Philip J. Guo. 2021. Ten Million Users and Ten Years Later: Python Tutor’s Design Guidelines for Building Scalable and Sustainable Research Software in Academia. In *The 34th ACM Symposium on User Interface Software and Technology*. 1235–1251. <https://doi.org/10.1145/3472749.3474819>
- [42] Philip J. Guo, Julia M. Markel, and Xiong Zhang. 2020. Learnersourcing at Scale to Overcome Expert Blind Spots for Introductory Programming: a Three-Year Deployment Study on the Python Tutor Website. In *Proceedings of the ACM Conference on Learning@ Scale (L@S 2020)*. 301–304. <https://doi.org/10.1145/3386527.3406733>
- [43] Nathan Hahn, Shamsi T. Iqbal, and Jaime Teevan. 2019. Casual Microtasking: Embedding Microtasks in Facebook. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2019)*. 1–9. <https://doi.org/10.1145/3290605.3300249>
- [44] Aaron Halfaker and R. Stuart Geiger. 2020. Ores: Lowering Barriers with Participatory Machine Learning in Wikipedia. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 1–37. <https://doi.org/10.1145/3415219>
- [45] Aaron Halfaker, R. Stuart Geiger, and Loren G. Terveen. 2014. Snuggle: Designing for Efficient Socialization and Ideological Critique. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2014)*. 311–320. <https://doi.org/10.1145/2556288.2557313>
- [46] Frances Haugen. 2021. *The Facebook Files: A Wall Street Journal Investigation*. <https://www.wsj.com/articles/the-facebook-files-11631713039>
- [47] Ting-Hao Kenneth Huang, Walter S. Lasecki, Amos Azaria, and Jeffrey P. Bigham. 2016. “Is There Anything Else I Can Help You With?” Challenges in Deploying an On-Demand Crowd-Powered Conversational Agent. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2016)*. <https://doi.org/10.48550/arXiv.1708.03044>
- [48] Shagun Jhaver, Iris Birman, Eric Gilbert, and Amy Bruckman. 2019. Human-Machine Collaboration for Content Regulation: The case of Reddit Automoderator. *ACM Transactions on Computer-Human Interaction (TOCHI)* 26, 5 (2019), 1–35. <https://doi.org/10.1145/3338243>
- [49] Jaejeung Kim, Chiwoo Cho, and Uichin Lee. 2017. Technology Supported Behavior Restriction for Mitigating Self-Interruptions in Multi-Device Environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1, 3 (2017), 1–21. <https://doi.org/10.1145/3130932>
- [50] Joongyim Kim, Taesik Gong, Kyungsik Han, Juho Kim, JeongGil Ko, and Sung-Ju Lee. 2020. Messaging beyond Texts with Real-Time Image Suggestions. In *Proceedings of the International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2020)*. 1–12. <https://doi.org/10.1145/3379503.3403553>
- [51] Tae Soo Kim, Seungsu Kim, Yoonseo Choi, and Juho Kim. 2021. Winder: Linking Speech and Visual Objects to Support Communication in Asynchronous Collaboration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2021)*. 1–17. <https://doi.org/10.1145/3411764.3445686>

- [52] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E. Kraut. 2011. Crowdforge: Crowdsourcing Complex Work. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2011)*. 43–52. <https://doi.org/10.1145/2047196.2047202>
- [53] Michal Kosinski, Sandra C. Matz, Samuel D. Gosling, Vesselin Popov, and David Stillwell. 2015. Facebook as a Research Tool for the Social Sciences: Opportunities, Challenges, Ethical Considerations, and Practical Guidelines. *American Psychologist* 70, 6 (2015), 543. <https://doi.org/doi/10.1037/a0039210>
- [54] Geza Kovacs, Zhengxuan Wu, and Michael S. Bernstein. 2018. Rotating Online Behavior Change Interventions Increases Effectiveness but also Increases Attrition. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–25. <https://doi.org/10.1145/3274364>
- [55] Adam D.I. Kramer, Jamie E. Guillory, and Jeffrey T. Hancock. 2014. Experimental Evidence of Massive-Scale Emotional Contagion through Social Networks. *Proceedings of the National Academy of Sciences* 111, 24 (2014), 8788–8790. <https://doi.org/10.1073/pnas.1320040111>
- [56] Daniel Lambton-Howard, Robert Anderson, Kyle Montague, Andrew Garbett, Shaun Hazeldine, Carlos Alvarez, John A. Sweeney, Patrick Olivier, Ahmed Kharrufa, and Tom Nappey. 2019. Whatfutures: Designing Large-Scale Engagements on WhatsApp. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2019)*. 1–14. <https://doi.org/10.1145/3290605.3300389>
- [57] Daniel Lambton-Howard, Emma Simpson, Kim Quimby, Ahmed Kharrufa, Heidi Hoi Ming Ng, Emma Foster, and Patrick Olivier. 2021. Blending into Everyday Life: Designing a Social Media-Based Peer Support System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2021)*. 1–14. <https://doi.org/10.1145/3411764.3445079>
- [58] Carlene Lebeuf, Margaret-Anne Storey, and Alexey Zagalsky. 2017. Software bots. *IEEE Software* 35, 1 (2017), 18–23. <https://doi.org/10.1109/MS.2017.4541027>
- [59] Yi-Chieh Lee, Naomi Yamashita, and Yun Huang. 2020. Designing a Chatbot as a Mediator for Promoting Deep Self-Disclosure to a Real Mental Health Professional. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (2020), 1–27. <https://doi.org/10.1145/3392836>
- [60] Qingyang Li, Clara Caldeira, Daniel A. Epstein, and Yunan Chen. 2020. Supporting Caring among Intergenerational Family Members through Family Fitness Tracking. In *Proceedings of the EAI International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth 2020)*. 1–10. <https://doi.org/10.1145/3421937.3422018>
- [61] Bin Lin, Alexey Zagalsky, Margaret-Anne Storey, and Alexander Serebrenik. 2016. Why Developers are Slackening Off: Understanding How Software Teams use Slack. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work and Social Computing Companion (CSCW 2016 Companion)*. 333–336. <https://doi.org/10.1145/2818052.2869117>
- [62] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. 2010. Turkit: Human Computation Algorithms on Mechanical Turk. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2010)*. 57–66. <https://doi.org/10.1145/1866029.1866040>
- [63] Fannie Liu. 2020. *Fostering Social Connection through Expressive Biosignals*. Ph.D. Dissertation. Carnegie Mellon University. <http://reports-archive.adm.cs.cmu.edu/anon/anon/home/ftp/usr0/ftp/hcii/CMU-HCII-20-103.pdf>
- [64] Fannie Liu, Laura Dabbish, and Geoff Kaufman. 2017. Supporting Social Interactions with an Expressive Heart Rate Sharing Application. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 1, 3 (2017), 1–26. <https://doi.org/10.1145/3130943>
- [65] Fannie Liu, Mario Esparza, Maria Pavlovskaia, Geoff Kaufman, Laura Dabbish, and Andrés Monroy-Hernández. 2019. Animo: Sharing Biosignals on a Smartwatch for Lightweight Social Connection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 3, 1 (2019), 1–19. <https://doi.org/10.1145/3314405>
- [66] Fannie Liu, Chunjong Park, Yu Jiang Tham, Tsung-Yu Tsai, Laura Dabbish, Geoff Kaufman, and Andrés Monroy-Hernández. 2021. Significant Otter: Understanding the Role of Biosignals in Communication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2021)*. 1–15. <https://doi.org/10.1145/3411764.3445200>
- [67] Lika Haizhou Liu, Xi Lu, Dennis Wang, Fannie Liu, Andrés Monroy-Hernández, and Daniel A. Epstein. 2022. Mindful Garden: Supporting Reflection on Biosignals in a Co-Located Augmented Reality Mindfulness Experience (*CSCW '22 Extended Abstracts*).
- [68] Miki Liu, Austin Wong, Ruhi Pudipeddi, Betty Hou, David Wang, and Gary Hsieh. 2018. ReactionBot: Exploring the Effects of Expression-Triggered Emoji in Text Messages. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–16. <https://doi.org/10.1145/3274379>
- [69] Zhicong Lu, Seongkook Heo, and Daniel J. Wigdor. 2018. Streamwiki: Enabling Viewers of Knowledge Sharing Live Streams to Collaboratively Generate Archival Documentation for Effective In-Stream and Post Hoc Learning. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–26. <https://doi.org/10.1145/3274381>
- [70] Zhicong Lu, Rubaiat Habib Kazi, Li-Yi Wei, Mira Dontcheva, and Karrie Karahalios. 2021. StreamSketch: Exploring Multi-Modal Interactions in Creative Live Streams. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1

- (2021), 1–26. <https://doi.org/10.1145/3449132>
- [71] Kai Lukoff, Taoxi Li, Yuan Zhuang, and Brian Y Lim. 2018. Tablechat: Mobile Food Journaling to Facilitate Family Support for Healthy Eating. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–28. <https://doi.org/10.1145/3274383>
- [72] Ulrik Lyngs, Kai Lukoff, Petr Slovak, William Seymour, Helena Webb, Marina Jirotnka, Jun Zhao, Max Van Kleek, and Nigel Shadbolt. 2020. ‘I Just Want to Hack Myself to Not Get Distracted’ Evaluating Design Interventions for Self-Control on Facebook. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2020)*. 1–15. <https://doi.org/10.1145/3313831.3376672>
- [73] Haley MacLeod, Ben Jelen, Annu Prabhakar, Lora Oehlberg, Katie A. Siek, and Kay Connelly. 2016. Asynchronous Remote Communities (ARC) for Researching Distributed Populations. In *Proceedings of the EAI International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth 2016)*. 1–8. <https://doi.org/doi/abs/10.5555/3021319.3021320>
- [74] Juan F. Maestre, Haley MacLeod, Ciabhan L. Connelly, Julia C. Dunbar, Jordan Beck, Katie A. Siek, and Patrick C. Shih. 2018. Defining through Expansion: Conducting Asynchronous Remote Communities (ARC) Research with Stigmatized Groups. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2018)*. 1–13. <https://doi.org/10.1145/3173574.3174131>
- [75] Kaitlin Maher, Amy X. Zhang, and David Karger. 2018. Squadbox: A Tool to Combat Email Harassment using Friendsourced Moderation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2018)*. 1–13. <https://doi.org/10.1145/3173574.3174160>
- [76] Jonathan T. Morgan, Siko Bouterse, Heather Walls, and Sarah Stierch. 2013. Tea and Sympathy: Crafting Positive New User Experiences on Wikipedia. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW 2013)*. 839–848. <https://doi.org/10.1145/2441776.2441871>
- [77] Sean A. Munson and Sunny Consolvo. 2012. Exploring Goal-Setting, Rewards, Self-Monitoring, and Sharing to Motivate Physical Activity. In *Proceedings of the Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth 2012)*. IEEE, 25–32. <https://doi.org/10.4108/icst.pervasivehealth.2012.248691>
- [78] Sean A. Munson, Erin Krupka, Caroline Richardson, and Paul Resnick. 2015. Effects of Public Commitments and Accountability in a Technology-Supported Physical Activity Intervention. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2015)*. 1135–1144. <https://doi.org/10.1145/2702123.2702524>
- [79] Sean A. Munson, Debra Lauterbach, Mark W. Newman, and Paul Resnick. 2010. Happier Together: Integrating a Wellness Application into a Social Network Site. In *Proceedings of the International Conference on Persuasive Technology (PERSUASIVE 2010)*. Springer, 27–39. https://doi.org/10.1007/978-3-642-13226-1_5
- [80] 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. <https://doi.org/10.1145/344949.344959>
- [81] Sneha Narayan, Jake Orlowitz, Jonathan Morgan, Benjamin Mako Hill, and Aaron Shaw. 2017. The Wikipedia Adventure: Field Evaluation of an Interactive Tutorial for New Users. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW 2017)*. 1785–1799. <https://doi.org/10.1145/2998181.2998307>
- [82] Michael Nebeling, Alexandra To, Anhong Guo, Adrian A. de Freitas, Jaime Teevan, Steven P. Dow, and Jeffrey P. Bigham. 2016. WearWrite: Crowd-assisted writing from smartwatches. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2016)*. 3834–3846. <https://doi.org/10.1145/2858036.2858169>
- [83] Midas Nouwens, Carla F. Griggio, and Wendy E. Mackay. 2017. “WhatsApp is for family; Messenger is for friends” Communication Places in App Ecosystems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2017)*. 727–735. <https://doi.org/10.1145/3025453.3025484>
- [84] Kathleen O’Leary, Stephen M. Schueller, Jacob O. Wobbrock, and Wanda Pratt. 2018. “Suddenly, we got to become therapists for each other” Designing Peer Support Chats for Mental Health. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2018)*. 1–14. <https://doi.org/10.1145/3173574.3173905>
- [85] Dan R. Olsen Jr. 2007. Evaluating User Interface Systems Research. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2007)*. 251–258. <https://doi.org/10.1145/1294211.1294256>
- [86] Annu Sible Prabhakar, Lucia Guerra-Reyes, Vanessa M Kleinschmidt, Ben Jelen, Haley MacLeod, Kay Connelly, and Katie A. Siek. 2017. Investigating the Suitability of the Asynchronous, Remote, Community-Based Method for Pregnant and New Mothers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2017)*. 4924–4934. <https://doi.org/10.1145/3025453.3025546>
- [87] Daniela Retelny, Sébastien Robaszkiewicz, Alexandra To, Walter S Lasecki, Jay Patel, Negar Rahmati, Tulsee Doshi, Melissa Valentine, and Michael S. Bernstein. 2014. Expert Crowdsourcing with Flash Teams. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2014)*. 75–85. <https://doi.org/10.1145/2642918.2647409>
- [88] Niloufar Salehi and Michael S. Bernstein. 2018. Hive: Collective Design through Network Rotation. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–26. <https://doi.org/10.1145/3274420>

- [89] Jomara Sandbulte, Chun-Hua Tsai, and John M. Carroll. 2021. Working Together in a PhamilySpace: Facilitating Collaboration on Healthy Behaviors Over Distance. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–32. <https://doi.org/10.1145/3449198>
- [90] Joseph Seering, Juan Pablo Flores, Saiph Savage, and Jessica Hammer. 2018. The Social Roles of Bots: Evaluating Impact of Bots on Discussions in Online Communities. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–29. <https://doi.org/10.1145/3274426>
- [91] Joseph Seering, Michal Luria, Connie Ye, Geoff Kaufman, and Jessica Hammer. 2020. It Takes a Village: Integrating an Adaptive Chatbot into an Online Gaming Community. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2020)*. 1–13. <https://doi.org/10.1145/3313831.3376708>
- [92] Katie A. Siek, Gillian R. Hayes, Mark W. Newman, and John C. Tang. 2014. Field deployments: Knowing from Using in Context. In *Ways of Knowing in HCI*. Springer, 119–142. https://doi.org/10.1007/978-1-4939-0378-6_6
- [93] Inc. Snap. 2020. *Can I Display Dynamic Information in a Lens?* <https://support.lensstudio.snapchat.com/hc/en-us/articles/360029378392-Can-I-display-dynamic-information-in-a-Lens->
- [94] Inc. Snap. 2020. *Creative Kit*. <https://kit.snapchat.com/docs/creative-kit>
- [95] Inc. Snap. 2020. *Lens Studio*. <https://lensstudio.snapchat.com/>
- [96] Hyewon Suh, John R. Porter, Alexis Hiniker, and Julie A. Kientz. 2014. @BabySteps: Design and Evaluation of a System for Using Twitter for Tracking Children’s Developmental Milestones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2279–2288. <https://doi.org/10.1145/2556288.2557386>
- [97] Katie Teague. 2020. *Gmail’s Scheduling Tool to Send Email Messages Later is One of Our Favorite Things*. <https://www.cnet.com/tech/mobile/gmails-scheduling-tool-to-send-email-messages-later-is-one-of-our-favorite-things/>
- [98] Jacob Thebault-Spieker, Stevie Chancellor, Michael Ann DeVito, Niloufar Salehi, Alex Leavitt, David Karger, and Katta Spiel. 2021. Do We Fix it or Burn it Down? Towards Practicable Critique at CSCW. In *Companion Publication of the ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW 2021 Companion)*. 234–237. <https://doi.org/10.1145/3462204.3483281>
- [99] John C. Thomas and Wendy A. Kellogg. 1989. Minimizing Ecological Gaps in Interface Design. *IEEE Software* 6, 1 (1989), 78–86. <https://doi.org/10.1109/52.16905>
- [100] Fernanda B. Viégas, Scott Golder, and Judith Donath. 2006. Visualizing Email Content: Portraying Relationships from Conversational Histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2006)*. 979–988. <https://doi.org/10.1145/1124772.1124919>
- [101] Dennis Wang, Marawin Chheang, Siyun Ji, Ryan Mohta, and Daniel A. Epstein. 2022. SnapPI: Understanding Everyday Use of Personal Informatics Data Stickers on Ephemeral Social Media. *Proceedings of the ACM on human-computer interaction* 7, CSCW. <https://doi.org/10.1145/3555652>
- [102] Steve Whittaker, Tara Matthews, Julian Cerruti, Hernan Badenes, and John Tang. 2011. Am I Wasting my Time Organizing Email? A Study of Email Refinding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2011)*. 3449–3458. <https://doi.org/10.1145/1978942.1979457>
- [103] Ting-Wei Wu, Yu-Ling Chien, Hao-Ping Lee, and Yung-Ju Chang. 2021. Im Receptivity and Presentation-Type Preferences among Users of a Mobile App with Automated Receptivity-Status Adjustment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2021)*. 1–14. <https://doi.org/10.1145/3411764.3445209>
- [104] Bin Xu, Pamara Chang, Christopher L. Welker, Natalya N. Bazarova, and Dan Cosley. 2016. Automatic Archiving versus Default Deletion: what Snapchat Tells Us about Ephemerality in Design. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work & Social Computing*. 1662–1675. <https://doi.org/10.1145/2818048.2819948>
- [105] Saelyne Yang, Changyoon Lee, Hijung Valentina Shin, and Juho Kim. 2020. Snapstream: Snapshot-Based Interaction in Live Streaming for Visual Art. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2020)*. 1–12. <https://doi.org/10.1145/3313831.3376390>
- [106] Amy X. Zhang and Justin Cranshaw. 2018. Making Sense of Group Chat through Collaborative tagging and Summarization. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–27. <https://doi.org/10.1145/3274465>
- [107] Amy X. Zhang, Grant Hugh, and Michael S. Bernstein. 2020. PolicyKit: Building Governance in Online Communities. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2020)*. 365–378. <https://doi.org/10.1145/3379337.3415858>
- [108] Amy X. Zhang, Andrés Monroy-Hernandez, and Michael Bernstein. 2021. *Social Computing Systems Summer Camp*. <https://socialcomputingsystemscamp.wordpress.com/>
- [109] Xuan Zhao, Cliff Lampe, and Nicole B. Ellison. 2016. The Social Media Ecology: User Perceptions, Strategies and Challenges. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2016)*. 89–100. <https://doi.org/10.1145/2858036.2858333>