# HINT: Integration Testing for AI-based features with Humans in the Loop

Quanze Chen*
cqz@cs.washington.edu
University of Washington
Seattle, WA, USA

Tobias Schnabel
Tobias.Schnabel@microsoft.com
Microsoft Research
Redmond, WA, USA

Besmira Nushi
Besmira.Nushi@microsoft.com
Microsoft Research
Redmond, WA, USA

Saleema Amershi
samershi@microsoft.com
Microsoft Research
Redmond, WA, USA

## ABSTRACT

The dynamic nature of AI technologies makes testing human-AI interaction and collaboration challenging – especially before such features are deployed in the wild. This presents a challenge for designers and AI practitioners as early feedback for iteration is often unavailable in the development phase. In this paper, we take inspiration from integration testing concepts in software development and present HINT (Human-AI INtegration Testing), a crowd-based framework for testing AI-based experiences integrated with a humans-in-the-loop workflow. HINT supports early testing of AI-based features within the context of realistic user tasks and makes use of successive sessions to simulate AI experiences that evolve over-time. Finally, it provides practitioners with reports to evaluate and compare aspects of these experiences.

Through a crowd-based study, we demonstrate the need for over-time testing where user behaviors evolve as they interact with an AI system. We also show that HINT is able to capture and reveal these distinct user behavior patterns across a variety of common AI performance modalities using two AI-based feature prototypes. We further evaluated HINT's potential to support practitioners' evaluation of human-AI interaction experiences pre-deployment through semi-structured interviews with 13 practitioners.

## CCS CONCEPTS

• **Human-centered computing** → **Usability testing**; *Collaborative and social computing*; *Interaction design process and methods*; *Human computer interaction (HCI)*.

## KEYWORDS

Human-AI interaction, prototyping, testing, crowdsourcing

*Work done as an intern researcher at Microsoft Research.


*IUI '22, March 22–25, 2022, Helsinki, Finland*

ACM ISBN 978-1-4503-9144-3/22/03…$15.00
https://doi.org/10.1145/3490099.3511141

**ACM Reference Format:**
Quanze Chen, Tobias Schnabel, Besmira Nushi, and Saleema Amershi. 2022. HINT: Integration Testing for AI-based features with Humans in the Loop. In *27th International Conference on Intelligent User Interfaces (IUI '22), March 22–25, 2022, Helsinki, Finland*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3490099.3511141

## 1 INTRODUCTION

Artificial Intelligence (AI) is deployed in many real-world applications such as email clients, text processing software or content curation platforms to augment human capabilities and catalyze productivity [52, 53]. Success of AI in these applications crucially hinges on successful Human-AI (HAI) collaboration. Fundamentally, human-AI collaboration is a dynamic experience – users adapt to the system as they repeatedly interact with it and build up experience. Likewise, the system can change and adapt to the user as it gathers more interaction data. Moreover, due to the statistical nature of AI models, failures in the predictions can happen unexpectedly at any time during the collaboration. Many of these failures in human-AI collaboration naturally require multiple interactions to manifest are thus difficult to detect. Unfortunately, isolated, offline testing of the AI which is an integral part of typical workflows is also insufficient – higher offline performance of the AI does not necessarily imply better collaboration outcomes [10, 15].

This increasingly presents a challenge to practioners working on AI-based features who need to evaluate the impact of changes to the AI's behavior on human collaborators. Current evaluation of AI-based features primarily centers around three main methods: (1) offline performance evaluation metrics based on test sets, (2) limited-scale user studies in the lab or (3) post-deployment A/B tests [27]. However, when considering the cost against the insights gained, these methods leave much to be desired.

Each evaluation method implicitly trades off costs to run (e.g., fidelity required, turnaround time, setup) with the costs to end-users (e.g., inadequate predictions) as well as the scale and richness (e.g., correlation with user satisfaction, granularity) of information gained. This is sketched out in Table 1. For example, offline tests of the AI alone give a statistically reliable estimate of the AI's accuracy (because of the typically large scale of offline datasets), require no working system to interact with and therefore have a low testing cost, and no cost implications to end-users. However, they also give little to no insight into how end-users experience

or adapt to the AI feature. Traditional A/B testing requires fully functional systems that can be deployed to end-users, presents high running costs (slow turnaround times) and can be costly to end-users who are experiencing failures, but they can measure user behavior at scale. Finally, lab-based user studies mitigate costs to end-users by using a small set of recruited participants, but are inherently costly to run. Lab-based user studies also trade off scale in favor of more in-depth feedback from a limited set of users. These limitations and tradeoffs of current techniques and the additional challenges of testing human-AI collaborative experiences often lead to unstructured ad-hoc testing or, in some cases, forgoing human participant tests altogether [52].

We propose HINT (Human-AI INtegration Testing), a customizable framework that addresses these drawbacks by allowing rapid testing of human-AI collaborative experiences. HINT takes inspiration from integration testing in software development [37] by automating testing of AI models *integrated* within an application and with humans-in-the-loop. HINT improves over offline testing by evaluating the AI with humans together in the context of the final application (system and interface) or application prototype. Additionally, HINT offers a simple way to initiate and scale these human participant tests by utilizing a crowdsourced test workflow.

HINT is designed to provide rich insights by capturing both observed user behavior and subjective metrics *over time*. HINT then distills its captured data into a structured test report that allows practitioners to answer critical questions that they could previously only answer post-deployment such as: *What might be the impact of a model update on my application's users? How might users react to improvements in my application's behavior due to personalization over time? How do my application's users react and recover from AI errors and what are the possible long-term consequences of those errors?*

To evaluate the usefulness of HINT, we created two prototypes of AI-based features for an email management application. We then simulated eight common scenarios motivated by the questions above and ran them through HINT. Our results show that HINT is able to surface distinct and evolving patterns of user behavior under these scenarios. Finally, through semi-structured interviews with real AI practitioners reviewing HINT generated reports, we demonstrate HINT's potential to support pre-deployment decisions that take into consideration evolving user experiences with AI systems.

In summary, this paper makes the following contributions:

- Human-AI INtegration Testing (HINT): a framework for rapid pre-deployment testing of AI-based features with humans-in-the-loop. The framework includes design of the testing setup, a crowdsourced workflow to execute tests, and a report summarizing information from user-centered (self-reported and measured) and offline metrics (→ Section 3).
- Results from an extensive set of feasibility studies demonstrating HINT's ability to capture and reveal distinct and evolving user behavior patterns. The studies use two AI-based feature prototypes tested across eight test scenarios with diverse over-time AI performance dynamics (→ Section 4).
- Results from semi-structured interviews with 13 AI practitioners reviewing HINT generated reports demonstrating HINT's potential to support pre-deployment evaluation of human-AI collaboration (→ Section 5).

## 2 RELATED WORK

Our work builds upon the long line of research on developing interactive, intelligent systems that can assist humans in performing complex tasks [19, 21]. AI-based features embedded in these systems introduce new challenges in the testing process [1, 2, 53, 55] due to (i) their probabilistic and non-deterministic nature, (ii) dynamic changes in system behavior and changes in user behavior, and (iii) higher likelihoods of being incorrect in practice because of training vs. real-world distribution discrepancies. These challenges motivate the HINT framework and the need for more human-centered tools and techniques for testing of AI in general. Next, we position our contributions in the context of traditional AI evaluation methods, UX/UI testing methods, and testing human-AI collaboration.

### 2.1 Evaluating the AI

A common initial step for testing an AI-based feature is evaluating the AI model itself as an independent component. Typically, this is done via offline evaluation where AI performance is measured on annotated ground truth datasets via a set of metrics (e.g., accuracy). As AI outputs have become more complex and context dependent, direct human evaluation of the AI system outputs is also increasingly common [13] and frameworks to efficiently conduct such evaluations such as MAISE [54] have been proposed in prior work. Decoupled (or component level) evaluation indeed provides a way for designers to quickly iterate on the AI itself.

However, decoupling the AI model from the actual application during testing presents only a limited view into the potential performance of the human and AI together as a collaborative system. In fact, prior work has recognized this issue of mismatched metrics and we will discuss them in more detail in Section 2.3.

Even putting these limitations aside, there are still major shortcomings in solely using offline evaluation. First, offline training/test datasets or human evaluation tasks are often based on a surrogate task (e.g., rating prediction) that the AI solves rather than the specific task (e.g., make good recommendations) and therefore can be misaligned to real-world user behavior [22]. Datasets also often contain artifacts of the annotation process used to create them [17], further biasing evaluations. Second, offline metrics (e.g., accuracy, F1, AUC, BLEU scores etc.) are only surrogate constructs and often fail to predict user-centric outcomes of interest (e.g., user satisfaction, intent to return) due to lack of expressivity in complex tasks [3, 10]. While human evaluation can resolve some of these issues, properly aggregating and understanding human evaluations can be a challenging task [50].

Finally, offline-only evaluation is not sensitive to time-related dynamics and cannot describe learning effects of either users adapting to the system and vice versa. While offline metrics evaluate over large sets of outputs, in reality a user will only experience a very small sample of the AI system's possible outputs over their sequential interactions. Good "average" performance on a test set does not matter if a user ends up encountering a few AI failures and proceeds to give up.

**Table 1: Conceptual space of testing techniques, illustrating different trade-offs between costs and informativeness of their results.**

| Method | System fidelity | Cost to run | Cost to users | Scale of data | Richness of data |
|---|---|---|---|---|---|
| Offline model tests | – | low | – | high | low |
| A/B testing | high | medium | high | high | low |
| In-lab user studies | medium | high | low | low | high |
| Prototype walkthroughs | low | low | low | low | high |
| HINT | medium/high | medium | low | medium | high |

One alternative to offline-only evaluation is A/B testing, which is already commonly used for understanding user experiences in both traditional and AI-based settings [16]. However, HINT and A/B testing focus on different phases in the development process: HINT testing supports more rapid iteration in the earlier phases of developing AI-based features by surfacing rich user interaction patterns without the long turnaround time and resource investment in building high-fidelity implementations necessary for A/B testing. A/B testing depends on natural user behavior which means collecting sufficient amounts of data to draw observations could involve long durations of waiting while HINT uses targeted tests scenarios set up by the practitioner to focus on a specific envisioned use. Pre-determined scenarios also allow practitioners to forgo instrumentation that would be necessary in A/B testing to understand the users' goals and intentions. While HINT and A/B testing share some similarities, HINT does not aim to replace traditional A/B testing that is done after development, but rather addresses the lack of efficient but also in-depth evaluation of the effects of over-time human-AI interaction during the development of AI-based features.

## 2.2 Evaluating the UX/UI

There is an abundance of prior work when it comes to understanding performance of user interfaces and interactions (UI/UX) [35] and one alternative is to treat the AI as a static aspect of the system and use usability evaluation methods such as prototype walkthroughs (low fidelity, pre-deployment) or in-lab user studies (medium to high fidelity, pre-deployment) to understand potential performance of human-AI collaborative features.

More recent work has also proposed a variety of systems and frameworks that combine crowdsourcing (to recruit participants) with remote usability testing [9, 11, 28, 31, 33, 57]. Studies have also been able to demonstrate the validatity of these crowdsourced testing methodologies by comparing them with parallel in-lab studies [28, 31]—while the authors observed methodological differences between the techniques, they found similar effect sizes and measurements, validating the effectiveness of the approach. Moreover, systems like CrowdStudy [33] have illustrated how to technically implement crowdsourced GUI tests for both large-screen and mobile interfaces.

Using crowdworkers to test applications and features driven by **fixed** AI model outputs has been proposed by prior work as well [42, 46]. However, with HINT we aim to address the need for understanding how users might adapt to a **dynamic** AI model, where the behavior of the underlying AI can evolve or change over time due to factors such as adaptation to users and model updates. As such, HINT builds upon the body of prior work on crowdsourced user testing to create a testing workflow that allows designers to test scenarios where underlying AI models also evolve and change over time. This fills a critical gap between offline evaluation of an AI model and evaluation of the UI/UX only with fixed AI model outputs.

## 2.3 Evaluating Human-AI Collaboration

Decoupled testing of AI-based systems has been widely recognized to be insufficient, especially when AI is deployed to partner with a human for problem-solving or decision-making tasks [4, 7, 32, 39]. In particular, experiments conducted in a collaboration context testify that there exist other dimensions that impact team accuracy rather than model accuracy alone:

- *Facilitation of justified trust* – The way how people *build and maintain trust* with an automated agent impacts collaboration [20, 30]. Therefore, AIs with similar accuracy but different affordances in helping users learn an accurate mental model of when and how the AI fails, result in very different team performance [4, 26]. A more fundamental aspect of justified trust is *confidence calibration* in predictive machine learning [34, 49] which has been empirically shown to improve human decision-making [23, 56]. Most relevant to our work, Bansal et al. [5] have shown that more accurate over-time updates can cause collaboration disruption due to the introduction of newly, unexpected errors. In these cases, updates that are less accurate but more consistent with the human mental model of trust may be preferred to maximize the accuracy of joint decision-making.
- *Interpretability* – Interpretability techniques have been extensively studied as a way of improving and justifying developer and/or user trust [8, 40]. However, numerous studies have identified issues when the AI model's predictions are made more interpretable, for example the risk of increasing user trust even when the prediction is wrong [6, 29, 38] or interpretation unfaithfulness to models [45].
- *Complementarity and human augmentation* – With AI performance continuing to improve, there exists the important question of whether these improvements also translate into improvements in overall performance as well [24, 32, 51]. In the next sections, we show how HINT can answer these questions by comparing the accuracy and effort of users without any AI assistance vs. with AI assistance from a system, and contrasting it with offline evaluation results. From a model training perspective, more recent

efforts have proposed to align model training and optimization such that it complements human expertise [32, 51].

We discuss the above human-centered dimensions to emphasize the potential discrepancy that may exist between model accuracy and its impact on users but that may be difficult to uncover unless AI applications are tested at a large scale, with HINT-like frameworks. Note that while the studies mentioned in this section actively involve users in the evaluation (unlike offline techniques), they do not evaluate the AI recommendations in the context of the actual system and GUI where it is integrated and deployed. HINT instead aims at making integration testing *in context* feasible before deployment.

## 3 DESIGN

The HINT framework organizes the testing process of AI-based systems into three high-level phases – **setting up** the AI feature to be tested, **conducting the tests** through a crowdsourced workflow, and **generating a report** based on the test data. As Figure 1 shows, HINT takes as input an AI-based system prototype, a user task definition and dataset and deploys it to the crowd. Crowd participants go through multiple task or calibration *sessions* while HINT is tracking user behavior and perceptions. Below are the details for each phase.

### 3.1 Setting Up HINT Tests

HINT groups individual tests into *scenario*s which are meant to simulate or replicate the dynamics of one possible way an AI may evolve or adapt over time. To set up HINT tests, some information needs to be configured by the tester to define the scenario. Specifically, this consists of the following three main components (cf. Figure 1):

(1) *GUI with AI-based feature* – This is a working, medium to high fidelity graphical interface crowd participants will interact with to solve the user task. The prototype only needs to contain functionality required to solving the user task (see below). The AI predictions may be pre-computed for easier deployment.
(2) *User task* – This defines a realistic task or use case that the AI-based feature is designed to support. A user task instance comprises a set of text definitions of what we expect the user may want to achieve, some criteria to determine how well the goal has been achieved, and the concrete input data involved in the task. For example, when testing a creativity support suggestion-based AI feature, a task may ask participants to create a set of slides based on a prompt and some assets. Individual sessions of this user task can then apply their own input data (e.g., a different prompt). If a performance metric/ground truth is available for the task, a tester can also provide it with the task to automatically evaluate participants' responses.
(3) *Workflow components* – This defines the dynamics of the user experience and AI-based feature through what we will refer to as *test blocks* (e.g., an individual task *session*) to run. Test blocks are organized into individual pages crowd participants go through and may be configured in a modular way. Test designers can flexibly choose from the following list of test blocks:
    - *Training* – familiarize the participant with the prototype and the task they will be solving in the form of a tutorial explaining how to use various parts of the system to solve the task.
    - *Calibration session* – ground the participant's expectations about the system using a baseline task. For example, calibration sessions can involve solving the task with the AI feature turned off. These are useful as baselines to understand how much value the AI may add to user productivity and can be used to calibrate for artifacts in results caused by recruitment variance.
    - *Task sessions* – present a task with some type of behavior dynamics applied for the AI. Interaction details are logged during these sessions via interaction probes (see next section).
    - *Task surveys* – collect self-reported feedback about the past session (task or calibration). Includes Likert-based qualitative probes for session as well as cumulative experience so far. Task surveys are matched to sessions and can only come after a session. The types of questions are configurable and we share our implementation of them in Section 3.2.1.
    - *Exit survey* – collects a final set of self-reported information from the user after they have completed all task sessions, supporting both free text as well as Likert rating or preference statements. The exit survey may be configured to include additional probe questions asking to compare experiences in sessions with AI-enabled features, against those without the AI. Test designers can choose what insights they want to capture and configure HINT accordingly.

    Task sessions, calibration sessions, and task surveys can be replicated multiple times when specifying a workflow which defines the length of the test. Training sessions and exit surveys should only be included once if at all and should be put exclusively at the beginning and the end of the workflow, respectively.

### 3.2 Crowdsourced Workflow for Data Collection

To collect user experiences in a scalable way, HINT runs the testing scenarios defined through the testing setup on a crowd platform (e.g., Amazon Mechanical Turk). This gives AI practitioners access to a large pool of recruitable participants allowing for easy scaling of testing.

Upon acceptance of the crowd platform task (HIT), participants will be debriefed about the test process and their consent will be collected. Following this, participants will go through the a series of *test blocks* as defined by the AI practitioner in the testing setup (Figure 1) until they complete all test blocks. For the test blocks involving task instances, HINT randomizes the available task instances to populate them, reducing any potential confounds resulting from the task data (e.g. slight differences in difficulty).

*3.2.1 HINT Probes.* HINT utilizes two type of probes to capture user behavior patterns: *measured metrics* (implicitly collected during task sessions) and *self-reported metrics* (explicitly collected during surveys). Below we describe the intent and nature of each probe type. The exact probes used may vary depending on the system being tested and we discuss how we implemented them for our experiments in Section 4.

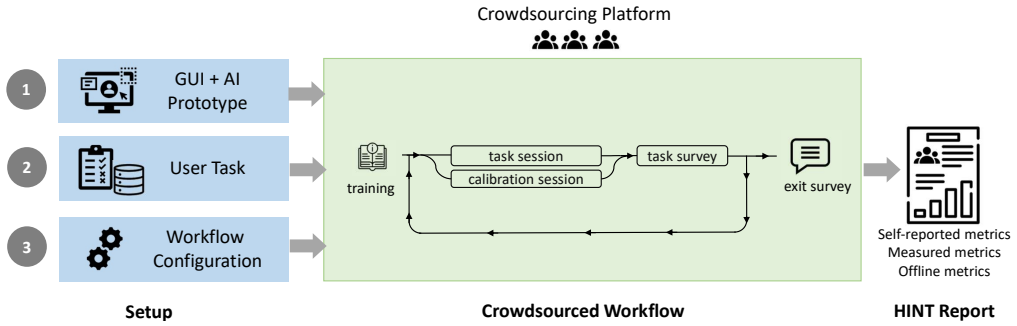



**Figure 1: High-level diagram of the HINT framework design. Tests consist of an AI-based feature, user task and the workflow configuration. HINT then recruits crowd participants to execute a customizable series of test blocks to probe changes in user behavior and perceptions over time (green). At the end, a report is produced to summarize the insights gained from testing.**

Measured metrics are collected by instrumenting the AI-based feature and are tied to the particular interactions in the interface.[1] For example, a probe might track button clicks in an AI-based feature that adds a button on suggested items or it may track text typed in a search query box (and the resulting items). HINT uses these implicit events to establish a timeline of user actions tied to each session. Metrics can then be inferred from the timelines by analyzing the events such as measuring their frequency or by finding specific sequences of events. These probes serve as analogs to what can be captured by traditional A/B testing but provide richer information due to their association with the session and their relation to other objective actions taken. Examples from different application domains include the number of opened emails for a mail client, the number of turns in a dialogue system, clickthrough rates in recommender systems, etc.

Self-reported metrics are collected through surveys at the end of a task session and are used to probe the user's qualitative evaluation of their experience. These are formatted in the form of Likert scores or open-ended feedback. Likert questions (on a 1-7 scale ranging from "strongly disagree" to "strongly agree") are used to probe statements mapping to experiences of interest. For example, in our validation experiments, we used the statement "*Based on my experience so far, the AI system can be trusted.*" as a probe for the self-reported trust metric.

## 3.3 Report

After data collection, HINT generates a report summarizing the data collected from crowd participants. A HINT report consists of five main sections: (1) the overview, (2) self-reported metrics (3) user voices (4) measured metrics and (5) correlations. Depending on the testing conducted, HINT reports may contain the results of a single standalone test or compare the results between two tests. An illustrative example of the report (comparing two tests) is shown in Figure 2. All graphs are interactive and magnifiable to allow for fine-grained insights.

The overview section (1) is designed to surface the most important high-level insights of the test. The report shows the over-time team performance (e.g., accuracy, F1 scores etc.) plotted along with corresponding offline AI performance for reference. It also shows a plot reflecting participants' effort over time. Depending on the application this can for instance be defined via number of actions, time spent in the task, or a combination of both. Finally, this section may also display participants' preference for the AI-based feature compared to the system without the AI-based feature if such a question was asked during the exit survey. If the report is comparative, it will show this information for both tests being investigated as additional lines or diverging bar charts.

In the self-reported metrics section (2), survey answers of participants' experience are aggregated and plotted for each of the HINT probes. When the report shows a single test, readers can also toggle the plotting behavior to view either mean Likert score evolution over time (via a line chart) or a detailed breakdown of Likert score distributions plotted as a diverging bar chart. To simplify information consumption, the report also provides linear fit coefficients on the left side to highlight increasing or decreasing trends over time.

At the end of the self-reported metrics section, the report presents samples of user voices (3) in the form of open-ended textual feedback. The goal of user voices is to show the participants' own assessment of how the system did or did not assist them and contextualize them when possible. When preferences are collected against a calibration (no AI) system, this feedback will be organized based on the participant's preference towards or against the AI-based feature.

In the section following, the report presents measured metrics (4) representing objective user behavior based on UI instrumentation. For each metric, linear fit coefficients are also shown.

Finally, at the end of the report (5), we provide a correlation table that shows the level of correlation between the measured metrics and the participants' self-reported metrics based on the collected data. High correlation between a self-reported and a measured metric can suggest potential proxies for tracking certain user experiences via implicitly tracked behaviors. This can enable practitioners to identify probes and events to instrument that may be useful in future A/B testing, as A/B testing crucially depends on finding metrics that are aligned with user outcome.

[1]Crowdsourcing workers are made aware and have consented to the instrumentation. Most importantly, these will not include any personal data since the data in the task is not tied to the worker.

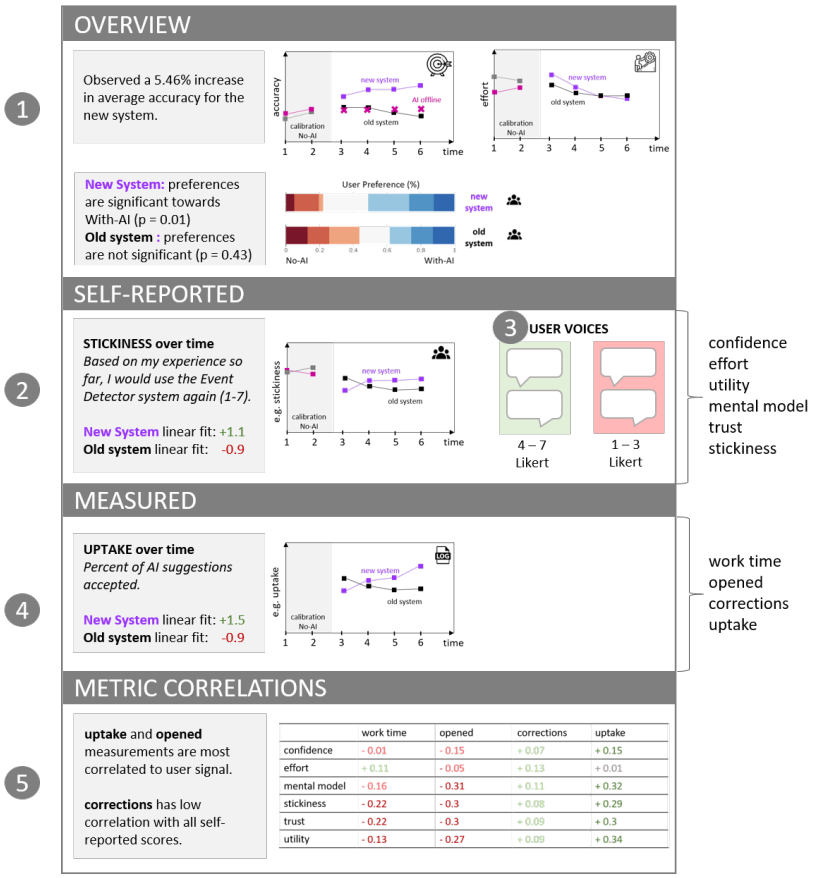



**Figure 2: Compact example of the HINT comparison report. This example outlines the main sections present in HINT reports and shows the visual form of information presented.**

## 4 EVALUATING THE HINT WORKFLOW

To evaluate the feasibility of our framework, we conducted HINT tests on two AI-based feature prototypes for an email management application. The tests span across different time-based variation schemes for the AI performance. Specifically, we set out to answer the following questions:

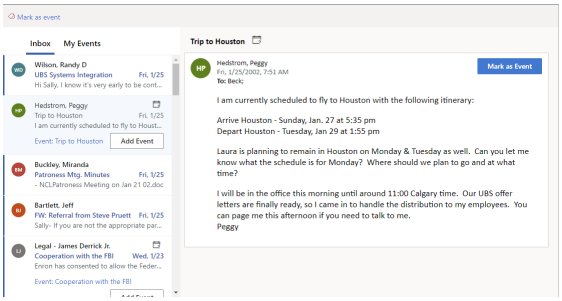
> **RQ1.** Are crowd workers able to use the HINT system to successfully complete tasks with the AI-based feature prototype?
> **RQ2.** Can the HINT framework reveal distinct patterns of user interactions under different performance variations over time?

In this section, we will first introduce the feature prototypes we used as examples tested using HINT, then we will present the setup and configuration for these tests. Finally, we will examine the insights revealed by HINT about these prototypes and how these insights address the research questions above.

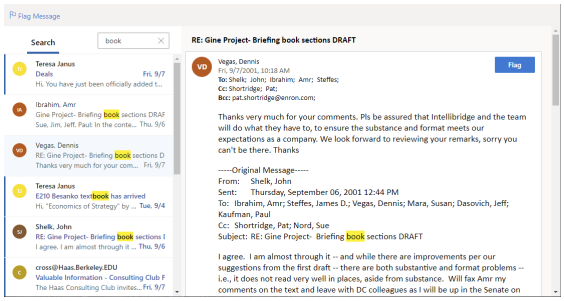
### 4.1 AI-based Feature Prototypes

For our experiments, we selected two AI-based features that support personal email management (interfaces shown in Figure 3). Personal email management is a domain that offers a rich space of tasks that can be aided by AI and past work [26] has explored such instances. Within the space of AI-based email management features, we decided on testing two common cases: AI-based search (SEARCH), implemented as an item ranking task, and AI-based event detection (EVENT), implemented as a classification task. For the AI assistance, we set up the search feature to AI to improve the text search results over the inbox by adjusting the ranking of search results to promote more relevant items (Figure 3b). For the event detection feature, the AI would assist by flagging messages (Figure 3a) containing events and showing an inlined "Add Event" button to allow the user to mark relevant messages.

*4.1.1 User tasks.* To ground these AI-based features for HINT testing, we also created appropriate tasks for each feature based on how each AI-based feature would be used. In the tests for the SEARCH feature, we designed the tasks to ask participants to find and tag a single email based on a description of the message (e.g., "*Find the email that contains the bill for the company offsite in August.*" ). In the tests for the EVENT feature, their task was to find and tag all emails that contained an event (e.g., meeting, appointment etc.).




(a) EVENT: Event detector marking emails classified as containing events with an action button.




(b) SEARCH: AI-based search ranking emails by relevance according to a query (here: "book").

**Figure 3: Examples of the email client prototypes with the two AI-based features.**

*4.1.2 Task data.* To populate data for each AI-based feature and task, we selected and edited messages from the Enron Email Dataset [25] to create a set of 12 inboxes (6 for SEARCH, 6 for EVENT). We set the size of the inboxes and ground truth solutions to control difficulty while simulating a realistic task. For the search case, we populated inboxes with 80 messages each out of which 1 message was the target message to be found. For the EVENT case, we selected a group of emails (ranging in size between 5-15) that contained event information for each inbox and populated the remaining emails with those that didn't contain information until the size of the inbox was between 35-45. The ground truth solutions for all the inboxes were selected by the research team from the Enron dataset such that all members agreed on the label of all messages in the inbox (with any ambiguous messages discarded).

## 4.2 Experiment Setup

In this section, we will breifly introduce the setup we used to run the HINT experiments including the probe metrics used, the *scenarios* we tested the AI-based feature prototypes on, and the recruitment of participants during the HINT tests.

*4.2.1 HINT Probes.* As described in Section 3.2.1, the general HINT framework allows for a variety of metrics to be captured as a part of the tests. For our experiments on the AI-based email management features, we used the following probe metrics:

- **Measured metrics:** Based on the nature of the application domain, we instrumented the prototype to track each user action in the context of the session (an activity log). Using the timeline, we derived the following measured (objective) metrics based on the log, noting that they represent only a subset of all potentially useful metrics:
  - **WORK TIME**: Time taken (s) from session start to complete.
  - **OPENED**: Number of messages opened. If a message was opened twice by a user, this would be counted twice.
  - **CORRECTIONS**: Ratio (%) of tagging actions that change a previously tagged item/email. A tagging action entails flagging for SEARCH and adding an event for EVENT.
  - **UPTAKE**: An overall metric for utilization of the AI-based feature. For EVENT, this is the % of tagging actions that were a result of accepting an AI suggestion. For SEARCH, this is the inverse of the number of queries made before an item was tagged.
- **Self-reported metrics:** Inspired by previous work on evaluating machine learning, human machine coordination and trust [14, 18, 47], we adapted and assembled six statements (formulated as 7-point Likert questions) for use as self-reported metrics in HINT. We drew three questions (confidence, effort, utility) from common questions used in user studies to evaluate general usefulness of features in the context of completing tasks. We then included questions about mental model and trust, which mirror similar questions used in the RADAR [47] evaluation but are adapted to guage *evolving* collaboration effectiveness between the human and AI by measuring the participant's cumulative trust and mental model of the AI so far as opposed to for a single session. Finally, we attempt to capture the overall quality of the collaboration over time by posing a question for whether the user will continue to use the AI system. The goal for this selection is to balance capturing key insights into human-AI teaming behavior, while maintaining a short and simple self-reporting process to reduce the cognitive load on the participants.
  - **CONFIDENCE**: "*I am confident that I completed the last task correctly.*"
  - **EFFORT**: "*Completing the task took me a lot of effort.*"
  - **UTILITY**: "*During the last task, the AI system was useful.*"
  - **MENTAL MODEL**: "*Based on my experience so far, I understand in what situations the AI system will perform well.*"
  - **TRUST**: "*Based on my experience so far, the AI system can be trusted.*"
  - **STICKINESS**: "*Based on my experience so far, I would use the AI system again.*"

  These questions generally fall into two groups: (1) questions about the last session (**CONFIDENCE**, **EFFORT**, **UTILITY**) and (2) questions about the cumulative (i.e., so far) experience (**MENTAL MODEL**, **TRUST**, **STICKINESS**).

*4.2.2 Workflow Setup.* For our experiments across all the conditions, we configured the HINT workflow with the same series of *workflow components*. Each participant was first given a *training* test block where they are given an example task to familiarize themselves with the basic interface of the email management feature.

Then each participant was assigned 6 sessions: 2 calibration sessions, followed by 4 task sessions. Task surveys were conducted after each session, followed by an exit survey at the end. The order of tasks used in the task sessions was randomized to account for any potential effects resuting from different task difficulty of the individual tasks used for each session.

*4.2.3 Scenario Conditions.* To simulate possible scenarios for the behavior of the AI, we created scenarios by applying one of 3 temporal variation schemes to one of the 2 AI-based features tested. These temporal variation schemes defined how the AI's performance would evolve over time during the 4 task sessions in that scenario. The three cases for temporal variation were: (1) **Static:** the AI has a static level of performance throughout all task sessions; (2) **Varies between sessions:** the AI's performance only changes between task sessions; (3) **Varies within session:** the AI's performance changes between different interactions within the same session. For each temporal variation scheme, there were two possible performance patterns based on a combination of sessions using a 'high-performance' AI (H) or a 'low-performance' AI (L). Overall, we created a total of 8 HINT tests, which are summarized in Table 2.

*4.2.4 AI performance for conditions.* In the **static** variation schemes, we simulated an AI system that had a fixed classification performance (precision and recall both at either 50% or 80% for the L and H respectively) throughout all the sessions. In the **varies between sessions** case, we used 2 types of patterns where we introduce a change in the performance level of the AI in the middle of the 4 sessions: HHLL (high then low) and LLHH (low then high). For the EVENT tests, the 'high-performance' AI (H) performed at 80% for both precision and recall while the the 'low-performance' AI (L) performed at 50% for precision and recall. In the SEARCH tests, we used a target mean reciprocal rank (MRR) to define the AI performance with 'high-performance' AI (H) defined as having a target MRR of 1 and the 'low-performance' AI (L) having a target MRR of 0.1.

Finally, for **varies within session** variation schemes, high variance (S-HV) and low variance (S-LV), the AI performance was targeted to hit an MRR value of either 1 or 0.1 (for each individual search query) in the high variance scheme, while the AI was targeted to consistently hit an MRR of 0.33 for the low variance scheme.

*4.2.5 Participants.* We deployed these tests on Amazon Mechanical Turk. Participants were given a base pay ($5.0) plus a per-session bonus of up to $0.5 based on their performance on the task to incentivize participants to make their best effort. Altogether, crowd workers received an average hourly wage at or above $12. We recruited 50 participants per test (out of 8 in Table 2) for a total of 400 participants. Our recruitment limited participants to participate in only one test per AI-based feature. For quality control, we discarded any participants who failed to find the target item in all 6 sessions for SEARCH feature tests and any participants who tagged less than 5 or more than 15 messages for all 6 sessions in the EVENT feature tests. We also discarded participants who did not complete all sessions. In total, we used data collected from 313 participants (E-L = 39 , E-H = 36, E-HHLL = 43, E-LLHH = 41, S-HHLL = 44, S-LLHH = 43, S-HV = 37, S-LV = 30).

## 4.3 Findings

*4.3.1 RQ1: Were crowd participants able to successfully complete tasks?* To evaluate **RQ1**, we probed participants' self-reported understanding of the tasks during each HINT test. We included the following 7-point Likert question: "*The task instructions were clear and easy to understand for all the task sessions.*" as part of the exit survey. We found that almost all participants reported agreement at any level (>4 Likert score) with this claim: 96% in the SEARCH tasks and 94% in the EVENT tasks. A significant majority also reported agreement or strong agreement (≥6 Likert score) in both EVENT (76.7%) and SEARCH (85.1%) tests.

Moreover, as evidenced by the performance in the calibration sessions (1 & 2), participants were able to achieve generally consistent and acceptable performance on the tasks even without AI support (Figures 5a,5c and 7a). This suggests that the HINT crowdsourced workflow for testing is adequate in allowing participants to understand the tasks in our test domains.

*4.3.2 RQ2: Can the HINT framework reveal distinct patterns of user interactions over time?* To evaluate **RQ2**, we looked at the collected probe metrics for pairs of performance variation schemes (as defined in Section 4.2.4). We were able to observe interesting patterns that were uniquely surfaced by the over-time interactions and measurements collected using the HINT framework.

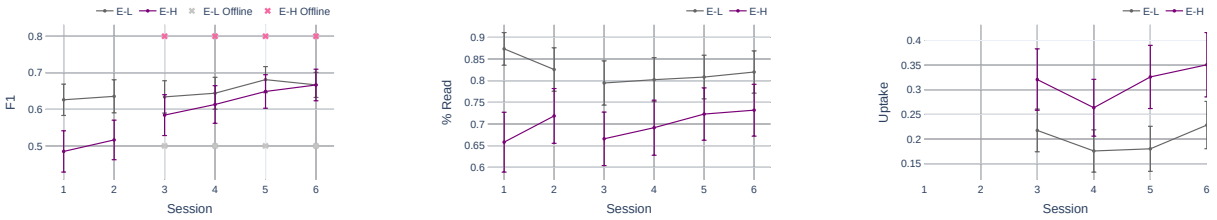
In the following sections, we will elaborate on the observations for each performance variation scheme.

*4.3.3 Static Performance – Figure 4.* The **static** performance tests simulate cases where the AI-based feature has consistent performance throughout its use. In our calibration sessions we observed that the E-H participants had significantly lower initial performance on the task without AI assistance than E-L participants (Figure 4a). This in conjunction with the difference in measured effort (percentage of emails opened, Figure 4b), shows that for these tests the pool of participants recruited differed in their abilites on the task. We suspect that this could be a rare case where attentive workers may have been exhausted in one test due to the simultaneous recruitment and in practice a practitioner should consider re-running additional test recruitments. While discrepancies in the participants base performance is undesirable, this observation does illustrate the importance of calibration sessions as anchor measurements in HINT to help us identify if any differences could be caused by artifacts due to the particular group of workers recruited.

Despite this difference in initial performance, we found that, surprisingly, in both AI performance patterns, participants ended up converging on similar levels of performance by the end of the last task session. In other words, the differential gain (change) in accuracy for the high-performance AI (E-H) when compared to No-AI sessions was higher than when using the lower-performance AI. At the same time, reliance on AI was also significantly higher for the E-H, seen through both higher uptake of the AI suggestions and lower amount of emails opened (Figure 4c and 4b). This indicated that despite having lower performance when doing the task without assistance, workers in this condition were able to discern that the AI performed well and accepted more recommendations from the AI than the group with a low-performance AI. It's worth noting that even though AI performance was static in these tests, we were

**Table 2: Summary description of tests conducted using the HINT workflow for the email management application.**

| Performance variation | Test | Feature | Description |
|---|---|---|---|
| **Static** | E-L (39) | EVENT | static performance, F1 = 0.5 |
| | E-H (36) | EVENT | static performance, F1 = 0.8 |
| **Varies between sessions** | E-HHLL (43) | EVENT | performance drops after 2nd AI session: F1 = 0.8 → 0.5 |
| | E-LLHH (41) | EVENT | performance increases after 2nd session: F1 = 0.5 → 0.8 |
| | S-HHLL (44) | SEARCH | performance drops after 2nd session: MRR = 1.0 → 0.1 |
| | S-LLHH (43) | SEARCH | performance increases after 2nd session: MRR = 0.1 → 1.0 |
| **Varies within session** | S-HV (37) | SEARCH | high performance variance within a session (random MRR target of 1.0 or 0.1 per query) |
| | S-LV (30) | SEARCH | low performance variance within a session (fixed MRR target of 0.33) |




**(a) F1 performance** **(b) Effort measured as % emails opened** **(c) Uptake measured as % tagged through AI suggestion**

**Figure 4: Over-time comparison of the E-L and E-H tests. Sessions 1, 2 are calibration sessions (no AI), while 3-6 are done with the AI. Effort is measured as % of all emails opened. Uptake is measured as % of all tagging completed through AI suggestion.**

only able to observe the convergence of the team performance and correlated differences in reliance on AI because of the over-time nature of the multiple sessions invloved in HINT tests (**RQ2**).

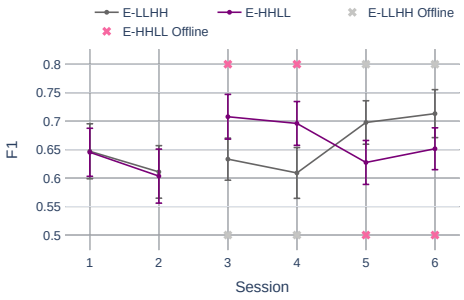
*4.3.4 Performance Varies Between Sessions – Figures 5 and 6.* In these tests, the performance of the AI was switched after the first 2 of the 4 AI sessions. In both AI-based features, HINT testing was able to surface the effects of this temporal change in AI performance reflected through changes in the overall team performance: We saw higher overall performance (Figure 5a and 5c) and higher uptake (Figure 5b and 5d) of the AI feature in the periods of time where the AI performed well.

Additionally, HINT revealed patterns in the user experiences that were unique to the particular AI-based feature being tested (SEARCH or EVENT). Looking the SEARCH tests, the temporal changes were reflected directly in both the reported and measured metrics: When AI performance dropped (HHLL) or increased (LLHH), we saw corresponding changes in probed values like uptake (Figure 5d) as well as self reported values like stickiness or effort (Figure 6b) with similar trends also present for confidence, trust, and mental model ratings. However, when we look at the EVENT tests, we only see a corresponding effect on these metrics (Figure 5b and 6a) when the performance dropped (HHLL) without seeing a similar change when the performance increases (llhh).

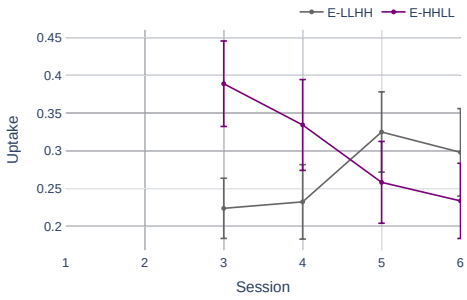
This seems to suggest that while participants were sensitive to AI performance changes in both directions when using the SEARCH feature, they were only sensitive to AI performance *drops* for the EVENT feature. An improvement in the AI performance did not result in a corresponding perceived difference by participants. We hypothesize that these effects may partially be explained by the different interactions involved in the two features – AI performance is easier to judge for the SEARCH feature because participants can quickly identify when the results ranks what they are seeking highly whereas better event tagging recommendations may just be ignored if the participant doesn't trust the feature anymore. This is also reflected by overall higher effort when comparing across the two tasks (effort in Figure 6a compared to 6b). Some of the user voices also support this hypothesis: A participant using the EVENT feature noted that "*It was wrong so often I just ignored it.*" while a participant using the SEARCH feature mentioned that "*It did not seem to be consistent. There was one task in particular where (...) the one I needed was the very last option. (...) Other times it worked seamlessly.*"

In this scenario, HINT testing was able to identify and surface a unique over-time behavior pattern resulting from the interaction between the AI's performance pattern and the task (SEARCH or EVENT). This is evidenced by clear over-time effects on measured (performance, uptake) and self-reported metrics, providing positive support for the sensitivity of HINT tests as well as the need for over-time testing (**RQ2**).

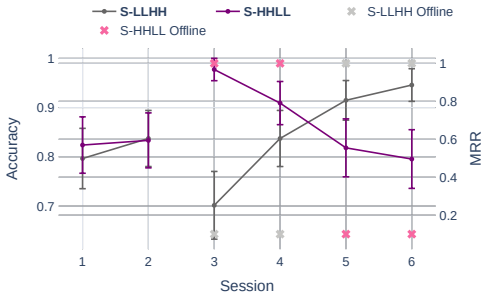
*4.3.5 Performance Varies Within Session – Figure 7.* Finally, we tested whether HINT is sufficiently sensitive to identify patterns induced by changes in AI performance within a single session –



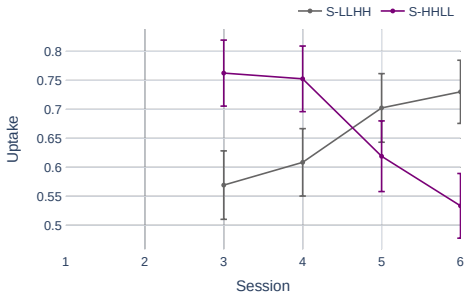

(a) F1 Performance Comparison




(b) Uptake as % tagged through AI suggestion

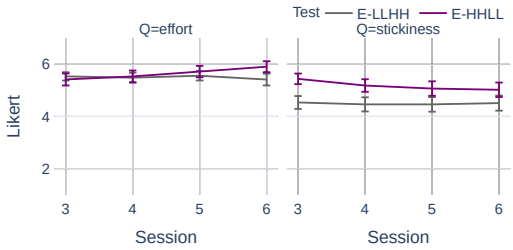



(c) Accuracy Comparison

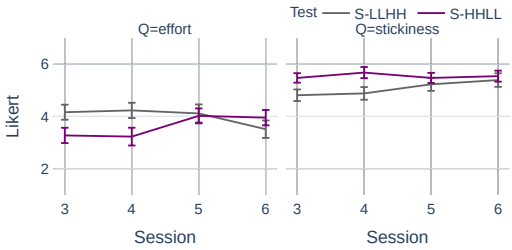



(d) Uptake as 1/queries before tagging

**Figure 5: Over-time comparison of performance and uptake for the EVENT (a, b) and SEARCH (c, d) features.**




**(a) Self-reported Likert Metrics for EVENT (E-LLHH, E-HHLL)**

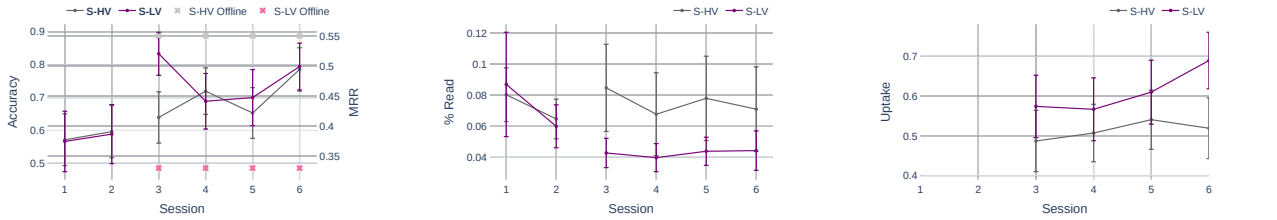



**(b) Self-reported Likert Metrics for SEARCH (S-LLHH, S-HHLL)**

**Figure 6: A subset of users' self-reported experience metrics for the EVENT feature (a) and the SEARCH feature (b) comparing the effect of changing AI performance from low to high v.s. high to low. Similar trends can be observed with the remaining features (See Figure 8 in the appendix)**

behavior that is more fine-grained than the probes used in HINT. We tested this with the SEARCH feature (conditions S-HV and S-LV). To recall, these tests compare a high-variance (but also high average performance) AI implementation that is sometimes high-performance while other times low-performance, with an AI implementation that is 'stable' between interactions with a lower average performance. Figure 7a shows the session-level performance with 'x' marks indicating average MRR performance for the AI feature in each condition.

First, we observed that the team performance (F1) across both tests mostly converged. This is in line with previous findings in information retrieval [48], showing that people are robust to retrieval quality, and consistent with our observations for a **static** AI. However, unlike with the **static** AI, we measured substantially higher effort (Figure 7b) for participants interacting with the high variance AI despite it having the higher average performance out of the two. Similarly, we observed a trend of increasing uptake for the low variance AI (despite its lower average performance) as opposed to a relatively flat uptake curve for the high variance AI (Figure 7c). We hypothesize that this difference would likely be attributed to the negative effects of unreliable performance (high-variance) outweighing the improvement in average performance.




(a) Accuracy Comparison (b) Effort measured as % emails opened (c) Uptake as 1/queries before tagging

**Figure 7: Over-time comparison of the S-HV and S-LV tests. Sessions 1, 2 are calibration sessions (no AI), while 3-6 are done with the AI. AI offline performance measured with average targeted Mean Reciprocal Rank (MRR). Uptake in SEARCH tasks is measured as queries$^{-1}$ issued before making a tag action.**

This is important because, in practice, it may not be evident to the practitioner ahead of time that there exists high variance in the AI-based feature unlike in our scenario. Offline tests based on fixed evaluation datasets can miss real-world variance while A/B tests in real-world situations can result in mixed signals as the users may naturally do tasks of differing difficulty that can be hard to compare. While the source of the unusual observations (compared to what would be expected of a **static** AI) in the previous paragraph can't be directly attributed to *variance* just by the HINT testing alone (since the probes don't measure interaction level variance), the existence of the behavior does suggest that HINT tests are able to uncover potentially anomalous user behavior patterns that can allow a practitioner to inform their future analysis. This further supports the case for HINT when it comes to revealing distinct behavior patterns (**RQ2**).

## 5 EVALUATING THE HINT REPORT

While our experiments in Section 4 evaluated the observations produced by the HINT framework, we also wanted to understand how HINT as a framework can provide utility to AI practitioners when it comes to informing their decision-making process. Specifically, we wanted to see whether the **report** produced by HINT allows practitioners to understand behaviors (especially over-time behaviors) of their AI system better, and whether this understanding would affect their decisions when it comes to deployment. To explore this, we conducted a qualitative evaluation by presenting practitioners with a deployment related scenario along with a HINT report based on the earlier experiments involving the AI-based email management feature prototypes.

In this section, we present qualitative results from our study that collected responses via semi-structured interviews from 13 AI practitioners at a large software company. We aimed at representing a balanced set of roles involved in the deployment decision process around AI features (program managers, designers, and engineers). Participants were recruited via three mailing lists focusing on the topics of "Machine Learning", "Design for AI", and "Machine Learning for Productivity". Participants' ages were diverse (min = 28, max = 53, mean = 41.7), as well their genders (6 male, 6 female, 1 prefer not to say). Table 3 shows the distribution of roles and years of experience in AI-based features for participants.

### 5.1 Study protocol

The one-hour long study started by presenting particpants with an overview of HINT, its overall goal and a description of the main terminology needed for the study as described in Section 3. Next, we described the AI-based feature that each participant would see (SEARCH or EVENT) and showed a screenshot of the medium-fidelity prototype of the email client to the participant, explaining the user task and how the AI-based feature would assist the user. After this, participants were asked to imagine they were part of a team working on the AI-based feature and need to make a deployment decision. Finally, we gave people a high-level overview of each section of the HINT reports along with quick explanation of what metrics they contained.

We assigned participants one of four different scenarios which different by the type of the deployment decision (See Appendix A). The set of scenarios was chosen to illustrate realistic use cases for HINT while also covering the different temporal variation schemes in the AI-based feature as well as different decision contexts (comparative vs. individual). Table 4 shows an overview of the properties of the scenarios we tested. With iterative changes being prevalent in development, 3 of our 4 scenarios assumed a comparative setting where two versions of the AI-based feature were being evaluated and 1 scenario covered the case where a single version of the AI-based feature was being evaluated.

We framed comparative deployment decisions (S1, S2, S4) as decisions between a simpler, less costly to run system versus a more costly system version with higher offline performance but unclear benefits. For example, scenario S1 was contextualized as follows:

> "*Imagine you have two AI system versions for the same AI-based feature for Event Detection. The old system has lower offline performance than the new system, but it would also be substantially cheaper to run as it does not require a complicated deep learning stack. You have already deployed the old system for a while and are now trying to decide on whether to switch to the new system. To make an informed decision, you run two tests via HINT: KEEPOLD simulating the case where you would not switch to the new system and OLDTONEW simulating the case where you would switch to the new system after the second session. Based on the HINT*

**Table 3: Distribution of roles and years of experience on AI-based features among AI practitioners in the study.**

| Role | Participants | Experience | Participants |
|---|---|---|---|
| Program Manager | 5 [P2, P18, P21, P32, P40] | < 1 year | 0 |
| Designer | 4 [P7, P9, P10, P23] | 1 - 2 years | 1 [P52] |
| ML Engineer / Scientist | 4 [P4, P20, P30, P36] | 2 - 5 years | 7 [P28, P29, P31, P37, P42, P51, P52] |
| | | > 5 years | 5 [P38, P40, P45, P45, P53] |

*report, would you advocate for keeping the old system or switching to the new one?*"

KeepOld and OldToNew are aliases for e-l and e-llhh, respectively. In scenario S3, we asked participants to decide whether a new AI-based feature was ready for deployment. Full descriptions of each can be found in the Appendix.

Participants were asked to think aloud while they were consuming the information in the report and deliberating while taking a deployment decision. After they made and justified their decision, we asked the following four open-ended questions in our interviews:

**Q1.** Would you use HINT for testing the AI-based features that your are working on?
**Q2.** How is the HINT report informing your decision-making?
**Q3.** What other types of tests would you run via HINT?
**Q4.** How can HINT reports be further strengthened to support different audiences and roles?

## 5.2 Summary of results

For each question, we grouped similar answers into themes and present them with a representative set of quotes.

*5.2.1 Q1. Would you use HINT for testing the AI-based features that your are working on?* Nine out of 13 participants in the study said that they would use HINT to test the features that they are working on. The main reasons for why participants would use HINT were being able to (i) access usage data with real users ahead of deployment, (ii) extend experimentation and testing beyond offline model evaluation, and (iii) select the metrics for A/B tests which are best correlated with end-user utility.

- P40 – Program Manager: "*Yes, I would definitely use something like this. One of the challenges that we have had is: 'How do we ensure that we experiment and have confidence way ahead of productization because productization is very resource heavy?'. So we need to get some early signal that it is great to actually transition out of the research phase [...] If we had a way to prototype and test these with customers early through something that we could crowdsource and if HINT did that for us, that would be extremely valuable because we need a way to test, get this data on real usage and real people and not just do offline model evaluation. We try to simulate this sometimes, but this isn't real users, it isn't qualitative feedback from real customers where they tell us that they are fighting with the system.*"
- P10 – Designer: "*At the very bottom [of the report], I really like this correlation table. If we do tests, and we determined what metrics were most useful proxies, that would be pretty cool. As this gets used, the authors would get metrics about where the system has given the wrong response. This is going to a level where we would do analysis with real users.*"

The four participants who were reluctant of using HINT mentioned that they would have challenges either integrating it with their current mode of operation, or that it is difficult to define a user task that encapsulates enough user context for testing. We discuss both these challanges in detail in Section 6.

- P21 – Program Manager: "*In our team, we need to think about the tradeoff between the hours spent on building the system vs. the cost that we would spend on human testing.*"
- P4 – Engineer: "*We often do the UX research upfront.*"
- P30 – ML Scientist: "*We run a recommendation service that requires quite a lot of personal context from the user. So in order to have an actual human judge for the effectiveness of the system, you have to be aware of that context.*"

*5.2.2 Q2. How is the HINT report informing your decision-making?* Overall, we saw the reports encouraged participants to think more carefully about how much value the AI-based features is adding for the user. We noticed this both in cases where the decision point was about deploying an AI on the first place (individual reports) or for replacing a current version with a more sophisticated one (comparative reports). For example:

- P32 – Program Manager: "*Because the user preference on AI is less than 40%, I think it is a little risky to roll it out right now. I would like to increase this to at least 60%-70% ...[gives an example where in their experience the AI feature was not a value added to the expert] ...In today's world, I think, expectations on AI have gone up.*"
- P20 - ML Engineer: "*The new system requires more engineering cost, support, and others. Since I don't see much clarity or strong signal on users preferring the new system more, then I am asking if it is really giving me enough value?*"

Participants also examined over-time behavior and trajectories when assessing systems:

- P7 – Designer: "*Now the time it takes [to complete the task] goes down which is great. You would hope that the amount of time it takes people to do the task goes down as they do it more.*"
- P9 – Designer: [Looking at overall performance] "*I think that over time, these two systems would perform similarily.*"

During the think-aloud sessions, we also observed participants switching between offline, self-reported, and measured metrics. Several reported that they appreciated the opportunity to join this type of information and correlate it in the last section of the report

**Table 4: Summary of scenarios and respective HINT reports shown to participants.**

| Scenario | AI-based feature | Participants | HINT report |
|---|---|---|---|
| S1 | EVENT | 4 [P2, P4, P30, P36] | comparative: E-L vs. E-LLHH |
| S2 | SEARCH | 3 [P7, P20, P21 | comparative: S-LV vs. S-HV |
| S3 | EVENT | 3 [P10, P32, P40] | individual: S-L |
| S4 | EVENT | 3 [P9, P18, P23] | comparative: E-L vs. E-H |

(e.g., P40 - PM: "*The table of metric correlation would definitely guide the metric focus for my experiments.*" ).

*5.2.3 Q3. What other types of tests would you run via HINT?.* During this part of the interview, participants mentioned other types of tests and open questions that they would like to answer prior to deployment. Extending test scenarios, participants mentioned the possibility of running N-way comparisons and reports or increasing the time horizon to understand how resilient a system is to absorb small changes over time. The other set of extensions targeted changes in the UX, sparking a variety of specific and broad questions around design, e.g.,

- P7 – Designer: [Talking about applying it to conversational AI] "*where you got two different personality styles for the AI [. . . ] and you test these different types of interaction.*"
- P40 – PM: "*Should I improve my model or improve the UX?*" )
- P2 – PM: "*What levers do I have to set the expectations right in UX design?*"

*5.2.4 Q4. How can HINT reports be further strengthened to support different audiences and roles?* Many participants expressed the need for different levels of granularity in the report (e.g., P7 - Designer: "*I don't think everyone wants the same level of information.*" ). On the low granularity end of the spectrum, one participant suggested it would be helpful to have a single score only (P4 - Engineer: "*From an engineer's POV – who is probably is downstream from the UX researcher – they would probably appreciate a single metric a bit better.*" ). Most suggestions targeted items of medium granularity – for example condensing a few scores into a summary score or adding a more general overview. However, participants also voiced concerns about high level summaries being effective (P9 – Designer: "*There is always a temptation to take the overview or like the more high level stuff and just run with it.*" ). We will return to this point in our discussion in the next section.

Further, participants mentioned that it would be helpful in some cases to guide interpretation of the graphs or recommend decisions:

- P32 – Program Manager: "*For each of the graphs, if you say this is good and this is bad (e.g. anything above X), that would also help in my decision making. You need to tell me what is good enough. . .* "
- P23 – Designer: "*Maybe there are best practices. [For example] if you are working on an e-mail system, here are the five things you want to be looking at.*"

# 6 DISCUSSION

Our results demonstrate that the HINT framework is practical and is able uncover complex over-time patterns in user behavior. We also make the case that these over-time patterns can arise in common situations. However, we also note that there are still limitations to the applicability of HINT and we will discuss those as well as potential avenues for future work in this section.

## 6.1 Limitations

One limitation present in our current experimental design is the use of novice crowd workers as the participants for our tests. It is not difficult to imagine cases where the AI-based feature is targeted towards domain experts (such as medical assistance tools or creativity support tools) that cannot be simulated by crowd workers. However, this limitation is mainly a feature of our experiments and not of the HINT framework itself. While some adjustments will need to be made when setting up aspects of the workflow – such as creating training sessions aimed at experts and reducing the frequency of user-reported metric probes – the overall process presented in the HINT workflow would be able to generalize to an expert-led setting.

However, there are still potential aspects that limit the applicability of HINT. For example, one key requirement for using HINT is that of being able to define appropriate tests – reminiscent of integration testing in software engineering. Defining an appropriate user task and producing the data to support that task will likely take some effort initially when running tests. This effort can likely be amortized over the number of future tests one will run but nonetheless is something to consider. Additional costs of testing may also be justified in some cases as passing tests can serve as certificates for key outcomes (e.g., increased human-AI performance). We recommend that teams working on AI-based features collectively decide on how many resources they want to dedicate to testing before designing HINT tests.

Additionally, some aspects of HINT and the report rely on being able to measure performance on a task in some objective way. This can present challenges in expert-led situations, such as evaluating creativity support tools, where there is no simple measure of performance. In these cases, the practitioner still has at their disposal the measured and self-reported probes in HINT and these measurements can still surface many valuable behavioral patterns even without performance-based anchors.

Finally, while HINT allows evaluating behavior patterns for specific user tasks and AI update scenarios, it doesn't by itself provide guidance for whether the task or design of the feature as defined by the practitioner is one that aligns best with the users' needs. Thus, we envision HINT to be used in conjunction with other tools and instrumentation at various stages of the design process.

### 6.2 Future Work

While we did not test HINT with highly personalized AI-based features, it would be interesting to run prototypes on each participant's own data, for example via authenticated service APIs. This also opens up the more general possibility of adapting HINT to more open-ended scenarios where tasks success is more subjective, e.g., in content recommendation. Another related issue is that of framing and motivating over time tasks for users. In our experiments, we framed it as a explicitly sequential experience – participants were asked to solve similar tasks repeatedly with the same AI and were asked about their experience *so far* after each session. However, other framings are possible, e.g,. asking the user to imagine they returned the next day to the AI-based feature in between sessions and asking whether they would like to use the AI again the next day. In a similar vein, it would be interesting to examine HINT workflows that would perform comparative tests explicitly, for example by enabling side-by-side comparisons or allow people to switch between different AI versions [12].

Regarding the HINT reports, we found a large interest in our interviews with practitioners in using them with their own AI-based features. While this is exciting, we envision multiple ways in which the report can further be strengthened or extended. An important direction to explore would be how to offer different levels of granularity in reporting those results. However, this needs to be carefully balanced with the goal of incentivizing practitioners to move away from single-score performance numbers since aggregated evaluation on single metrics may hide important conditions of failure [5, 36, 41]. Therefore, we expect future iterations on the HINT report design to balance between aggregated and detailed interaction, and most importantly to allow for interactive data filtering and progressive information presentation [43]. Zooming in could add support for being able to replay individual user traces or for understanding anomalies or outliers. Another ask from practitioners was to add more active guidance or decision support to the reports. This may take the form of automatically generated insights or recommending a certain deployment decision. Future work could therefore consider extending HINT into a decision-support system, keeping in mind that this may cause undesired over-reliance on the system due to automation bias [44].

## 7 CONCLUSION

In this work we presented HINT, a customizable crowd-based framework for testing human-AI collaboration over time. HINT supports scalable testing of medium to high fidelity prototypes with crowdworkers performing real tasks over successive sessions and generates summary reports that enable practitioners to see over time trends in human-AI collaboration that they previously could only see post-deployment. We found that in common scenarios that involve an evolving AI-based feature, there were distinct behavior patterns that were only surfaced by tests that focused on performance variations over time. Our evaluations show that HINT is sensitive enough to reveal these behavioral trends resulting from the evolving experience between humans and AI systems and that HINT reports can provide early signals to help practitioners make deployment decisions that consider the effects of AI systems on the user experience. In summary, we propose the HINT framework to address a major gap in tools and methodologies for testing human-AI interaction experiences.

## REFERENCES


[1] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 291–300.

[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N Bennett, Kori Inkpen, et al. 2019. Guidelines for human-AI interaction. In *Proceedings of the 2019 chi conference on human factors in computing systems*. 1–13.

[3] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*. Springer, 382–398.

[4] Gagan Bansal, Besmira Nushi, Ece Kamar, Walter S Lasecki, Daniel S Weld, and Eric Horvitz. 2019. Beyond accuracy: The role of mental models in human-AI team performance. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 7. 2–11.

[5] Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S Weld, Walter S Lasecki, and Eric Horvitz. 2019. Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 2429–2437.

[6] Gagan Bansal, Tongshuang Wu, Joyce Zhu, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel S Weld. 2020. Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance. *arXiv preprint arXiv:2006.14779* (2020).

[7] Zana Buçinca, Phoebe Lin, Krzysztof Z Gajos, and Elena L Glassman. 2020. Proxy tasks and subjective measures can be misleading in evaluating explainable AI systems. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*. 454–464.

[8] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1721–1730.

[9] Yan Chen, Maulishree Pandey, Jean Y. Song, Walter S. Lasecki, and Steve Oney. 2020. Improving Crowd-Supported GUI Testing with Structural Guidance. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376835

[10] Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2020. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review* (2020), 1–56.

[11] Eelco Dolstra, Raynor Vliegendhart, and Johan Pouwelse. 2013. Crowdsourcing gui tests. In *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*. IEEE, 332–341.

[12] Michael D Ekstrand, Daniel Kluver, F Maxwell Harper, and Joseph A Konstan. 2015. Letting users choose recommender algorithms: An experimental study. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 11–18.

[13] Markus Freitag, George F. Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation. *ArXiv* abs/2104.14478 (2021).

[14] Susan R. Fussell, Robert E. Kraut, F. Javier Lerch, William L. Scherlis, Matthew M. McNally, and Jonathan J. Cadiz. 1998. Coordination, Overload and Team Performance: Effects of Team Communication Strategies. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work* (Seattle, Washington, USA) *(CSCW '98)*. Association for Computing Machinery, New York, NY, USA, 275–284. https://doi.org/10.1145/289444.289502

[15] Florent Garcin, Boi Faltings, Olivier Donatsch, Ayar Alazzawi, Christophe Bruttin, and Amr Huber. 2014. Offline and online evaluation of news recommender systems at swissinfo. ch. In *Proceedings of the 8th ACM Conference on Recommender systems*. 169–176.

[16] Juan Cruz Gardey and Alejandra Garrido. 2020. User Experience Evaluation through Automatic A/B Testing *(IUI '20)*. Association for Computing Machinery, New York, NY, USA, 25–26. https://doi.org/10.1145/3379336.3381514

[17] Mor Geva, Y. Goldberg, and Jonathan Berant. 2019. Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets. *ArXiv* abs/1908.07898 (2019).

[18] Alyssa Glass, Deborah L. McGuinness, and Michael Wolverton. 2008. Toward Establishing Trust in Adaptive Agents. In *Proceedings of the 13th International Conference on Intelligent User Interfaces* (Gran Canaria, Spain) *(IUI '08)*. Association for Computing Machinery, New York, NY, USA, 227–236. https://doi.org/10.1145/1378773.1378804

[19] William E Hefley and Dianne Murray. 1993. Intelligent user interfaces. In *Proceedings of the 1st international conference on Intelligent user interfaces*. 3–10.

[20] Kevin Anthony Hoff and Masooda Bashir. 2015. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human factors* 57, 3 (2015), 407–434.

[21] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 159–166.

[22] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)* 25, 2 (2007), 7–es.

[23] Matthew Kay, Tara Kola, Jessica R Hullman, and Sean A Munson. 2016. When (ish) is my bus? user-centered visualizations of uncertainty in everyday, mobile predictive systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 5092–5103.

[24] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. 2018. Human decisions and machine predictions. *The quarterly journal of economics* 133, 1 (2018), 237–293.

[25] Bryan Klimt and Yiming Yang. 2004. The Enron Corpus: A New Dataset for Email Classification Research. In *ECML*. 217–226.

[26] Rafal Kocielnik, Saleema Amershi, and Paul N Bennett. 2019. Will you accept an imperfect ai? exploring designs for adjusting end-user expectations of ai systems. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14.

[27] Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M Henne. 2009. Controlled experiments on the web: survey and practical guide. *Data mining and knowledge discovery* 18, 1 (2009), 140–181.

[28] Steven Komarov, Katharina Reinecke, and Krzysztof Z Gajos. 2013. Crowdsourcing performance evaluations of user interfaces. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 207–216.

[29] Himabindu Lakkaraju and Osbert Bastani. 2020. " How do I fool you?" Manipulating User Trust via Misleading Black Box Explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 79–85.

[30] John D Lee and Katrina A See. 2004. Trust in automation: Designing for appropriate reliance. *Human factors* 46, 1 (2004), 50–80.

[31] Di Liu, Randolph G Bias, Matthew Lease, and Rebecca Kuipers. 2012. Crowdsourcing for usability testing. *Proceedings of the American Society for Information Science and Technology* 49, 1 (2012), 1–10.

[32] Reid McIlroy-Young, Siddhartha Sen, Jon Kleinberg, and Ashton Anderson. 2020. Aligning Superhuman AI and Human Behavior: Chess as a Model System. *arXiv preprint arXiv:2006.01855* (2020).

[33] Michael Nebeling, Maximilian Speicher, and Moira C Norrie. 2013. Crowdstudy: General toolkit for crowdsourced evaluation of web interfaces. In *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*. 255–264.

[34] Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*. 625–632.

[35] Jakob Nielsen. 1994. *Usability engineering*. Morgan Kaufmann.

[36] Besmira Nushi, Ece Kamar, and Eric Horvitz. 2018. Towards Accountable AI: Hybrid Human-Machine Analyses for Characterizing System Failure. In *Proceedings of the Sixth AAAI Conference on Human Computation and Crowdsourcing*. 126–135.

[37] M. Ould and Charles Unwin. 1987. Testing in software development. *The Mathematical Gazette* 71 (1987), 331–331.

[38] Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. 2018. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810* (2018).

[39] Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. 2019. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220* (2019).

[40] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[41] Marco Túlio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. 4902–4912.

[42] Tobias Schnabel, Gonzalo Ramos, and Saleema Amershi. 2020. "Who Doesn't like Dinosaurs?" Finding and Eliciting Richer Preferences for Recommendation. In *Fourteenth ACM Conference on Recommender Systems* (Virtual Event, Brazil) *(RecSys '20)*. Association for Computing Machinery, New York, NY, USA, 398–407. https://doi.org/10.1145/3383313.3412267

[43] Ben Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE symposium on visual languages*. 336–343.

[44] Linda J Skitka, Kathleen L Mosier, and Mark Burdick. 1999. Does automation bias decision-making? *International Journal of Human-Computer Studies* 51, 5 (1999), 991–1006.

[45] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. 2019. How can we fool LIME and SHAP? Adversarial Attacks on Post hoc Explanation Methods. *arXiv preprint arXiv:1911.02508* (2019).

[46] Arjun Srinivasan, Mira Dontcheva, Eytan Adar, and Seth Walker. 2019. Discovering Natural Language Commands in Multimodal Interfaces. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, California) *(IUI '19)*. Association for Computing Machinery, New York, NY, USA, 661–672. https://doi.org/10.1145/3301275.3302292

[47] Aaron Steinfeld, Rachael Bennett, Kyle Cunningham, Matt Lahut, Pablo-Alejandro Quinones, Django Wexler, Daniel Siewiorek, Paul Cohen, Julie Fitzgerald, Othar Hansson, Jordan Hayes, Mike Pool, and Mark Drummond. 2006. *The RADAR Test Methodology: Evaluating a Multi-Task Machine Learning System with Humans in the Loop*. Technical Report CMU-CS-06-125. Carnegie Mellon University, Pittsburgh, PA.

[48] Andrew H Turpin and William Hersh. 2001. Why batch and user evaluations do not give the same results. In *SIGIR*.

[49] Ben Van Calster and Andrew J Vickers. 2015. Calibration of risk prediction models: impact on decision-analytic performance. *Medical decision making* 35, 2 (2015), 162–169.

[50] Chris Welty, Praveen K. Paritosh, and Lora Aroyo. 2019. Metrology for AI: From Benchmarks to Instruments. *ArXiv* abs/1911.01875 (2019).

[51] Bryan Wilder, Eric Horvitz, and Ece Kamar. 2020. Learning to Complement Humans. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. 1526–1533. https://doi.org/10.24963/ijcai.2020/212

[52] Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. 2018. Investigating How Experienced UX Designers Effectively Work with Machine Learning. In *Proceedings of the 2018 Designing Interactive Systems Conference* (Hong Kong, China) *(DIS '18)*. Association for Computing Machinery, New York, NY, USA, 585–596. https://doi.org/10.1145/3196709.3196730

[53] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. In *Proceedings of the 2020 chi conference on human factors in computing systems*. 1–13.

[54] Omar Zaidan. 2011. MAISE: A Flexible, Configurable, Extensible Open Source Package for Mass AI System Evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Edinburgh, Scotland, 130–134. https://aclanthology.org/W11-2114

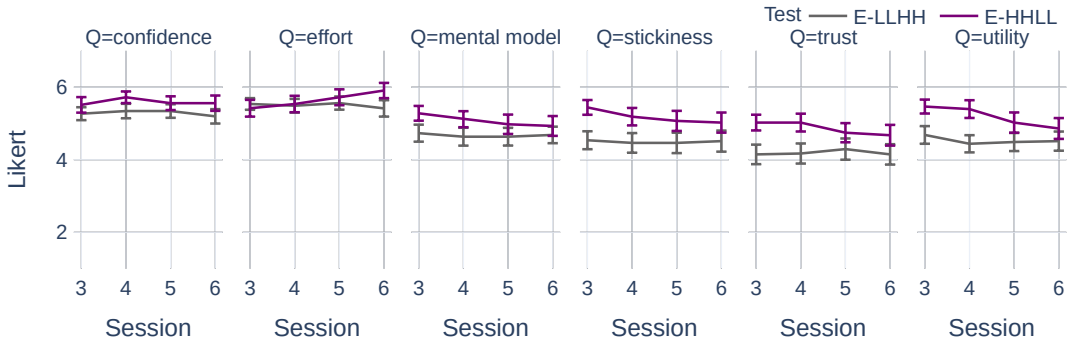
[55] Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. 2020. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering* (2020).

[56] Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 295–305.

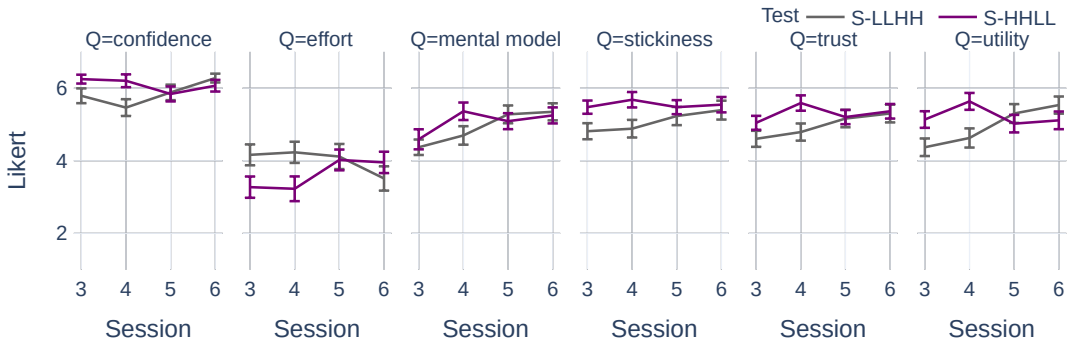
[57] Shkodran Zogaj, Ulrich Bretschneider, and Jan Marco Leimeister. 2014. Managing crowdsourced software testing: a case study based insight on the challenges of a crowdsourcing intermediary. *Journal of Business Economics* 84, 3 (2014), 375–405.

# A APPENDIX

- **S1 – Event Detection**: "*Imagine you have two AI system versions for the same AI-based feature for Event Detection. The old system has lower offline performance than the new system, but it would also be substantially cheaper to run as it does not require a complicated deep learning stack. You have already deployed the old system for a while and are now trying to decide on whether to switch to the new system. To make an informed decision, you run two tests via HINT: KeepOld simulating the case where you would not switch to the new system and OldToNew simulating the case where you would switch to the new system after the second session. Based on the HINT report, would you advocate for keeping the old system or switching to the new one?*"
- **S2 – Search**: "*Imagine you have a simple baseline system for an AI-based feature, in this case email Search. However, this baseline system does not have state-of-the art accuracy. You are thinking about deploying a newer AI system which in overall has higher accuracy than the previous baseline, but you believe is rather inconsistent across interactions. To take an informed decision, you run two tests via HINT: S-LV (to simulate the case when you deploy the simple baseline) and S-HV (to simulate the case when you would deploy the more accurate but less consistent system). Based on the HINT report, would you advocate for deploying the simple baseline or the newer AI system? ”*
- **S3 – Event Detection**: "*Imagine you have a simple baseline system for an AI-based feature, in this case Event Detection. However, this baseline system does not have state-of-the art accuracy and you would like to understand whether it adds anything over a simple no-AI system and see how new users are adapting to it. To take an informed decision, you run a test via HINT: S-LLLL (to simulate the deployment of the simple baseline). Based on the HINT report, would you advocate for deploying the AI-based feature with the simple baseline?*"
- **S4 – Event Detection**: "*Imagine you have two AI system versions for the same AI-based feature, in this case Event Detection. One system is a simple baseline which is fast to run but does not have state-of-the art accuracy. The other system is more sophisticated and more accurate offline but also more computationally expensive (requires a complicated deep learning stack). You would like to understand whether there exists a marginal user benefit of one system over the other. None of these systems has been deployed in the past in production. To take an informed decision, you run two tests via HINT: E-LLLL (to simulate the deployment of the baseline system) and E-HHHH (to simulate the deployment of the more sophisticated system). Based on the HINT report, would you advocate for deploying the AI-based feature with the simple baseline or the sophisticated system? ”*




**(a) Self-reported Likert Metrics for EVENT (E-LLHH, E-HHLL)**




**(b) Self-reported Likert Metrics for SEARCH (S-LLHH, S-HHLL)**

**Figure 8: Full plots of all probes for users' self-reported experience metrics for the EVENT feature (a) and the SEARCH feature (b) comparing the effect of changing AI performance from low to high v.s. high to low.**