VisTA: Integrating Machine Intelligence with **Vis**ualization to Support the Investigation of **T**hink-**A**loud Sessions

Mingming Fan, Ke Wu, Jian Zhao, Yue Li, Winter Wei, and Khai N. Truong

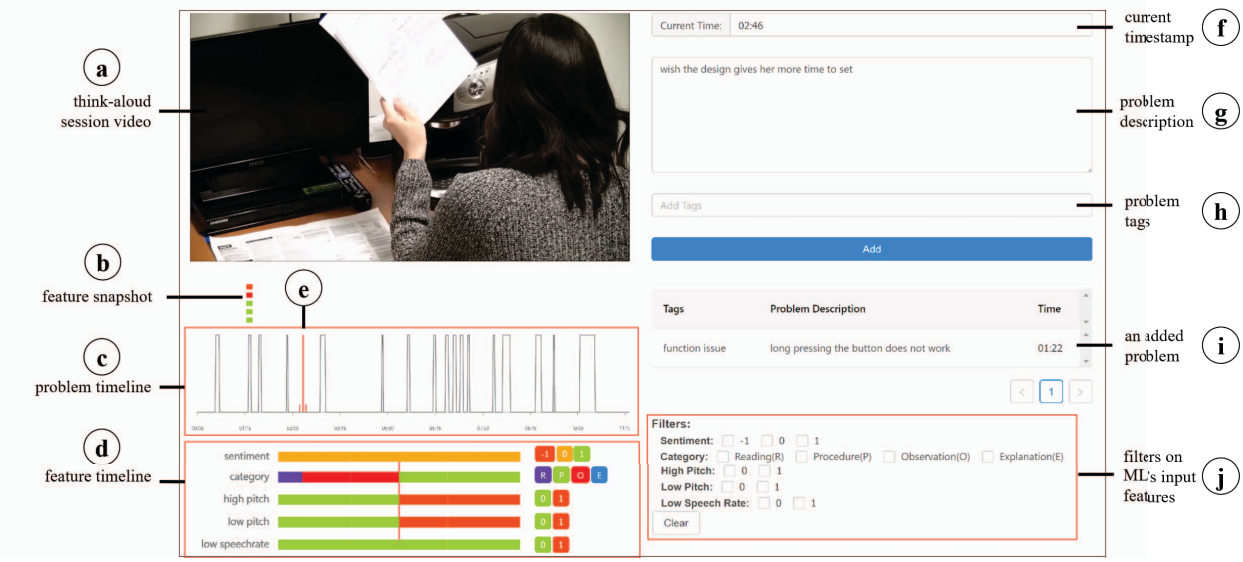

Fig. 1: VisTA: a visual analytics tool that allows UX practitioners to analyze recorded think-aloud sessions with the help of machine intelligence to detect usability problems.

Abstract—Think-aloud protocols are widely used by user experience (UX) practitioners in usability testing to uncover issues in user interface design. It is often arduous to analyze large amounts of recorded think-aloud sessions and few UX practitioners have an opportunity to get a second perspective during their analysis due to time and resource constraints. Inspired by the recent research that shows subtle verbalization and speech patterns tend to occur when users encounter usability problems, we take the first step to design and evaluate an intelligent visual analytics tool that leverages such patterns to identify usability problem encounters and present them to UX practitioners to assist their analysis. We first conducted and recorded think-aloud sessions, and then extracted textual and acoustic features from the recordings and trained machine learning (ML) models to detect problem encounters. Next, we iteratively designed and developed a visual analytics tool, *VisTA*, which enables dynamic investigation of think-aloud sessions with a timeline visualization of ML predictions and input features. We conducted a between-subjects laboratory study to compare three conditions, i.e., *VisTA*, *VisTASimple* (no visualization of the ML's input features), and *Baseline* (no ML information at all), with 30 UX professionals. The findings show that UX professionals identified more problem encounters when using VisTA than Baseline by leveraging the problem visualization as an overview, anticipations, and anchors as well as the feature visualization as a means to understand what ML considers and omits. Our findings also provide insights into how they treated ML, dealt with (dis)agreement with ML, and reviewed the videos (i.e., play, pause, and rewind).

Index Terms—Think-aloud, visual analytics, machine intelligence, user study, usability problems, session review behavior, UX practices

1 Introduction

Think-aloud protocols were initially developed in psychology to study people's thought processes when solving problems [13] and were later introduced into the human-computer interaction (HCI) field to identify usability problems with interface design [27]. It is considered as the single most useful usability testing method [35] and often used by the majority of user experience (UX) practitioners in usability testing [31].

Although it is beneficial to conduct many rounds of usability testing in the early stage of a project [28], analyzing testing sessions can be time-consuming and UX practitioners often work under time pressure to deliver results in time [7, 15]. Recent research has shown that when users encounter problems in think-aloud sessions, their verbalizations tend to include more observations, negative sentiments, questions, abnormal pitches, and speech rates [14]. Thus, there is an opportunity to leverage the patterns to increase the efficiency of UX practitioners in analyzing large amounts of think-aloud sessions. At the same time, however, we face many open questions.

First, with the advancement in natural language processing (NLP) and machine learning (ML), it is interesting to explore whether ML models can be designed to detect where in a recorded think-aloud

- *Mingming Fan is with University of Toronto and Rochester Institute of Technology. Email: mfan@cs.toronto.edu.*
- *Ke Wu, Yue Li, Winter Wei, and Khai N. Truong are with University of Toronto. E-mails: {kedaniel.wu, shurrik.li, winter.wei}@mail.utoronto.ca, khai@cs.toronto.edu*
- *Jian Zhao is with University of Waterloo. E-mail: jianzhao@uwaterloo.ca. Work was completed while the author was at FXPAL.*
- *Mingming Fan is the corresponding author. Ke Wu and Jian Zhao contributed equally. The work is supported by NSERC.*

Manuscript received 31 Mar. 2019; accepted 1 Aug. 2019.
Date of publication 16 Aug. 2019; date of current version 20 Oct. 2019.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2019.2934797

session the user likely encounters usability problems. Second, for better utilization of the prediction, visualizations can be designed to present the usability problem encounters (referred as *problems* hereafter) detected by ML and enable effective exploration. Would the visualization improve UX practitioners' performance or offer them a different perspective to scrutinize usability problems? Third, since recent research shows that many UX practitioners have little experience working with an artificial intelligence (AI) agent or understand the capabilities and limitations of ML [10,41,42], how would they perceive and manage their relationship with ML during their analysis?

In this research, we take the first step toward answering these questions by designing and evaluating a visual analytics tool powered by ML to assist UX practitioners with interactively investigating video-recorded think-aloud sessions.

We first conducted and recorded think-aloud sessions in which eight participants used both digital and physical products. We transcribed the sessions, labeled user-encountered problems as the ground truth, classified verbalizations into categories [8, 12], and automatically extracted textual and acoustic features. We used these information to train a range of ML models, including random forest (RF), support vector machine (SVM), convolutional neural network (CNN), and recurrent neural network (RNN), to detect usability problems that users encountered, and evaluated the model performances.

Following an iterative user-centered design process, we developed *VisTA*, a visual analytics tool that allows UX practitioners to interactively explore and analyze recorded think-aloud sessions with machine intelligence (Fig. 1). The tool presents the ML-inferred problems along a timeline, as well as the verbalization and speech features that the ML model takes as input, allowing for a better understanding of the model. In addition, VisTA provides a video player for browsing recorded sessions, and offers the capabilities of annotating and tagging identified problems.

To deeply understand how UX practitioners perceive and utilize ML-inferred problems with this visual analytics approach, we conducted a between-subjects controlled study with 30 UX practitioners. In addition to *VisTA*, we included a *Baseline* condition, in which UX practitioners did not have the assistance from ML, to learn about the impact of ML. We also included a *VisTASimple* condition, in which UX practitioners were only able to see the ML predictions (without showing input features), to evaluate the effect of having access to these verbalization and speech features in their usage of ML.

In sum, our contributions in this paper are in two-fold:

- A novel visual analytics tool, VisTA, to assist UX practitioners with analyzing recorded think-aloud sessions, which integrates ML for usability problem detection with interactive visualization of ML-inferred usability problems and the ML's input features as well as video review and problem annotation functions;
- Results of a controlled user study that quantitatively and qualitatively characterize how UX practitioners used the tool and provide in-depth insights into how they leveraged and perceived ML in their analysis and how they reviewed think-aloud videos.

2 Background and Related Work

2.1 Think-aloud Verbalizations and Usability Problems

Think aloud protocol was developed in psychology to study human thought processes [13]. Later it was introduced into the HCI field to understand usability problems [27]. McDonald et al. surveyed the use of the think aloud protocol and found that 90% of the UX practitioners in their study often use it [31] and it was considered as the single most valuable usability engineering method [34].

When using think aloud, participants verbalize their thought processes while carrying out a task. Participants' verbalizations provide data to understand their thought processes. To safeguard the validity of the verbalizations, Ericcson and Simon and a later meta analysis suggest three guidelines: use a neutral instruction to think aloud that does not request specific types of verbalizations; use a think-aloud practice session to allow participants to become familiar with verbalizing their thoughts; use a neutral "keep talkin" token to remind participants to think aloud if they fall into silence [13, 16]. In practice, usability professionals may not adhere to the three guidelines [5]. For instance, usability evaluators may probe participants, which can cause changes in their behavior [13, 22]. Thus, we followed the three guidelines to conduct think-aloud sessions in this research.

Researchers systematically decomposed users' verbalizations into more manageable segments and categorized them into four categories: *Reading (R)*—read words, phrases, or sentences directly from the device or instructions; *Procedure (P)*—describe his/her current/future actions; *Observation (O)*—make remarks about the product, instructions or themselves; and *Explanation (E)*—explain motivations for their behavior [8]. This categorization was further validated by Elling et al. [12] and was often cited by later work [21, 24, 32, 45]. Recently, Fan et al. studied the verbalization categories and speech features (e.g., pitch, speech rate) of segments in which users encountered problems and found that when users experienced problems, their verbalizations tend to include the *Observation* category, *negative sentiment*, *negations*, *questions*, and *abnormal pitches* and *speech rates* [14]. Inspired by this finding, we extend this line of research by examining if it is possible to automatically detect problems based on these verbalization and speech features that tend to occur when users encountered problems; and how to integrate such machine intelligence into visual analytics to assist UX practitioners to better analyze think-aloud sessions.

2.2 Machine Learning for User Experience (UX)

As ML is increasingly integrated into products, it is inevitable that UX practitioners would encounter ML in their workflow. However, a recent survey revealed that many UX practitioners struggled to understand the capabilities and limitations of ML and they often tend to join projects after functional decisions have been made [10]. Even if UX practitioners can join projects early, they often fail to see places where ML could improve UX [42]. Consequently, many UX practitioners are unprepared to effectively leverage ML capabilities [10, 41, 42] where it may be able to enhance user experience.

To address this problem, some developed education materials that aim to teach UX practitioners technical concepts of ML [19]. Some organized workshops to bring designers and technologists together to explore how ML might function as a creative material [17]. These work implies that designers should gain technical knowledge of ML. However, a recent interview study with UX practitioners, who had years of experience designing ML-enhanced products, found that they knew little about how ML works but yet they still could use their "designerly abstraction" to work with ML [41]. This finding supports that it is possible for designers to treat ML as "design material" when improving UX with it [40]. Inspired by this idea, we would like to understand how UX practitioners would use, perceive, and react to ML by creating an opportunity for them to work with ML when analyze think-aloud sessions and learn from their experiences.

2.3 Visual Analytics to Facilitate Qualitative Research

Purely applying machine learning to solve qualitative research tasks can be challenging. That is because machine learning models are often used to classify or cluster data into categories, but qualitative researchers might need more than automatically generated labels. Further, the results generated by machine learning models may be inaccurate. Researchers attempt to integrate human knowledge and machine intelligence via interactive visualization. For example, Drouhard et al. designed a tool called Aeonium to facilitate collaborative qualitative coding process [11]. Aeonium highlights ambiguity in qualitative coding and facilitates the evolution of code definitions.

Moreover, several visualization systems have been proposed to analyze interaction logs, which helps qualitative researchers recover users' intentions and reasoning processes behind. The interaction logs can include low-level inputs (e.g., mouse clicks and eye-tracking data) and high-level actions (e.g., zooming and panning). Heer et al. discussed the design space of interaction histories and proposed a thumbnail-based approach for showing the graph structures of user interactions [20]. The HARVEST system aims to capture the provenance of users' insights based on their low-level inputs [18]. Interaction logs and think-aloud data are used together to help

users recall their strategies in visual analytics systems [9, 30]. In addition, various visualization techniques have been proposed for displaying eye-tracking data, including point-based, AOI-based, and those integrating both. Blascheck et al. provided a comprehensive survey on this topic [3]. By combining think-aloud, interaction, and eye movement data together, VA^2 facilitates the analysis of multiple concurrent evaluation results via coordinated views [2]. Also, researchers have investigated user-generated annotations and developed visual interfaces to assist with the discovery of higher-level patterns in users' sense-making processes [43, 44].

While several previous tools utilize think-aloud data for analyzing users' behaviors [2, 9, 30], unlike VisTA, verbalization and speech patterns have not been explored to build ML that detects problems and assists UX practitioners with analyzing think-aloud sessions. Further, it is still an open question how UX practitioners would perceive and work with ML during their analysis. In this paper, we strive to address this by conducting user studies to compare different conditions of integrating ML into the visual analytics system.

3 Research Questions

In this work, we explore the following research questions (RQs) to understand how UX practitioners would work with ML in a visualization to analyze video-recorded think-aloud sessions:

RQ1: *How would UX practitioners leverage ML in their analysis?* How would they use the visualization of ML? Would it help them identify more problems?

RQ2: *How would ML influence UX practitioners' session review strategies?* If so, how many types of strategies are there? Would they tend to review session recordings with more rewinds or pauses?

RQ3: *How would UX practitioners perceive and manage the relationship with ML?* What are their attitudes? How would they deal with the agreement, disagreement, and limitations of ML?

We iteratively developed a visual analytics tool, VisTA, that integrates machine intelligence and used it as a vehicle to answer the RQs. We designed a controlled study to expose UX practitioners to ML at different levels, and recorded a rich set of quantitative and qualitative data about their interactions with the tool, their analysis behaviors (e.g., pauses and rewinds), and their perceived relationship with ML.

In the rest of paper, we first describe how we curated a dataset of think-aloud sessions and trained ML models to detect problems (Sec. 4) and how we designed the visual analytics tool (Sec. 5). Next, we explain our study design and analysis methods (Sec. 6). Finally, we present the results and discuss our findings (Sec. 7 & 8).

4 Think-Aloud Dataset and Problem Detection

4.1 Data Collection

To curate a think-aloud dataset, we recruited 8 native English speakers (4 females and 4 males, aged 19–26) to participate in our think-aloud study. Participants had diverse education backgrounds including humanity, engineering, and sciecnes.

In our think-aloud study, we collected data of participants using three different interfaces, including one digital product (i.e., a science and technology museum website) and two physical products (i.e., a universal remote control and a multi-function coffee machine).

During the study, the moderator first played a short video tutorial [36] to demonstrate how to think aloud, and then asked each participant to practice thinking aloud when setting an alarm on an alarm clock. Next, each participant used each of the three products (in a counter-balanced order) to complete a task while thinking aloud. The tasks were related to major functions of the products and were as follows: 1) search for a photo of the instructions for an early telescope, 2) program the coffee machine to make two cups of strong-flavored drip coffee at 7:30 in the morning, and 3) program a remote control to operate a DVD player. For physical products, participants were also given a hard-copy of their instruction manuals.

All think-aloud sessions were video recorded with audio stream. The average session duration was 222 seconds ($\sigma = 131$) for the website, 619 seconds ($\sigma = 195$) for the universal remote control, and 854 seconds ($\sigma = 251$) for the coffee machine.

Table 1: The performance of the ML models trained with input features.

	TF-IDF/ Word embedding			All features		
	Precision	Recall	F-score	Precision	Recall	F-score
RF	0.79	0.53	0.64	0.80	0.64	0.71
SVM	0.59	0.73	0.65	0.76	0.70	0.73
CNN	0.81	0.41	0.54	0.79	0.48	0.60
RNN	0.60	0.43	0.50	0.76	0.54	0.64

4.2 Data Labeling and Feature Extraction

The think-aloud sessions were manually transcribed into text. Then, two coders divided each think-aloud session recording into small segments, similar to the approach in [8, 12]. The beginning and end of a segment was determined by pauses between verbalizations and the verbalization content [8, 12]. Each segment corresponded to a verbalization unit, which could include single words, but also clauses, phrases and sentences. For each segment, two coders first labeled independently whether the think-aloud user encountered a problem (e.g., being frustrated, confused or experiencing a difficulty) and later discussed to consolidate their labels. We used the binary problem labels as the *ground truth* for training ML models. In total, there were 3080 segments, of which users encountered problems in 370 segments.

For each segment, two coders assigned it with one of the four verbalization categories (i.e., reading, procedure, observation, and explanation) [8]. Recent research found when users encounter problems in think-aloud sessions, their verbalizations tend to include the *Observation* category, *negative sentiment*, *negations*, *questions*, and *abnormal pitches* and *speech rates* [14]. Inspired by this finding, in addition to labeling the *category* information for each segment, we computed its *sentiment* based on the transcript using the VADER library [25]. Moreover, we designed a keyword matching algorithm to determine whether users verbalized *negations* (e.g., no, not, never) in a segment. Similarly, we designed a keyword matching algorithm to determine whether users *asked a question* in a segment by searching for keywords (e.g., what, when, where) that were located at the beginning of a sentence. Lastly, for each segment, we computed user's *pitch* (HZ) using the speech process toolkit Praat [4] and the *speech rate* by dividing the number of words spoken in a segment by its duration. To determine whether the user verbalized with abnormally high or low pitches and speech rates, we computed the mean and the standard deviation (STD) of the pitch and the speech rate of the entire think-aloud session and automatically labeled a segment as having *abnormally high or low pitch or speech rate)* if any value in the segment was two standard deviations higher or lower than the mean pitch or speech rate.

In sum, six *verbalization* features were generated for each segment: category, sentiment, negations, questions, abnormal pitches, and abnormal speech rates. In addition, for each segment, we also computed the *TF-IDF* (i.e., term frequency-inverse document frequency) using scikit-learn library [37] and trained *word embeddings* on our dataset using Tensorflow [1]. In the end, eight features were used as the input for training a range of machine learning models to determine whether the user encountered a problem in each segment.

4.3 Model Training and Evaluation

We employed four machine learning methods: random forest (RF), support vector machine (SVM), convolutional neural network (CNN), and recurrent neural network (RNN), which have been shown effective in text-based classification tasks.

We extracted the TF-IDF features for each segment in the dataset and used them to train SVM and RF models using the scikit-learn. We used the word-embedding features to train CNN and RNN models. The CNN had an embedding layer followed by a convolution layer, a ReLu layer, a max pooling layer, and then a softmax layer. The RNN had an embedding layer followed by an LSTM with GRU (Gated Recurrent Unit) as the RNN cell and softmax as the activation function. To evaluate the models, We performed a 10-fold cross validation on our data set and used the performance of these models as the *baseline*. In addition, we appended the verbalization and speech features (i.e., category, sentiment, negations, questions, abnormal

Fig. 2: An early version of VisTA that visualizes the verbalization and speech features of the entire think-aloud session to UX evaluators.

pitches, and abnormal speech rates) to the end of the TF-IDF or word embedding vector for each segment as the input to train the same four ML models. Similarly, we performed 10-fold cross validation.

The results in Table 1 show that verbalization and speech features helped improve all ML models' performance. The SVM models performed the best. CNN and RNN did not outperform SVM or RF probably because our dataset was relatively small for CNN or RNN to learn optimal hyper parameters . Thus, we decided to use the best performed SVM models to predict the problem labels (i.e., whether the user encountered a problem or not) for all segments of the think-aloud sessions. After this process, all segments in each think-aloud session had a binary ML-inferred problem label.

5 VisTA: *Vis*ual Analytics for *T*hink-*A*loud

Following a typical user-centered iterative design process, we developed VisTA that interactively presents the verbalization features and the problems detected by ML as described earlier.

5.1 Design Principles

Our initial design of VisTA presents the verbalization and speech features of the entire think-aloud session to UX practitioners as a series of synchronized timelines (Fig. 2), in addition to the functions that allow evaluators to play the recorded sessions and add problem descriptions. To get a sense of the effectiveness of this design, we recruited 12 UX practitioners (8 females and 4 males, aged 22–31) as usability evaluators and asked them to use the tool to analyze the recorded think-aloud sessions. Afterwards, we interviewed them to understand their usage and preferences of the tool functionalities. Each study session lasted about 1.5 hours and each evaluator was compensated with $30. Based on the results, we derived two principles to improve the design of VisTA.

- *Be Simple and Informative.* Evaluators wanted to have a simple interface that offers concise information that they could consult to if need, while allowing them to focus on watching or listening to the recorded sessions. Although evaluators felt that each of the feature can be informative, showing all of them at once was overwhelming, as one evaluator pointed out that *"because lines are so busy, it is hard to pick up significant areas while reviewing the session."* Instead of viewing all the raw features and trying to figure out important information, they would prefer just having one *condensed* type of information while still being able to access the raw features if needed.
- *Be Interactive and Responsive.* Evaluators felt that the function of clicking anywhere on any timeline to move the session recording to that timestamp was helpful. In addition, they wanted to interact with the input features, such as filtering particular features, to better understand and leverage the features. Evaluators also wanted to tag their identified problems with short annotations to facilitate their analysis.

Fig. 3: The *feature timeline* shows ML's main input features in a short time window around the current time and updates as the video plays.

We adopted these two principles in the redesign of VisTA. Specifically, we integrated the machine intelligence into the analysis flow among other capabilities (Fig. 1). The refined VisTA interface provides a typical video player, a *problem timeline* that visualizes the ML-inferred problems, and a *feature timeline* that visualizes ML's input features on the left, as well as a panel on the right for logging and tagging identified problems and filtering input features.

5.2 Session Reviewing and Problem Logging

A UX evaluator can play and pause the think-aloud video (Fig. 1a) by pressing the ESC key, or fast-forward or backward by pressing the right or left arrow keys. While the video is playing, a red vertical line (Fig. 1e) moves to indicate the current timestamp, which is also automatically updated (Fig. 1f). Evaluators can write a problem description (Fig. 1g), add short and reusable problem tags (Fig. 1h), and finally log the problem by pressing the "Add" button.

All problems that the evaluator identified are visualized in the problem table (Fig. 1i). Clicking a problem entry in the table navigates the video to the timestamp on the timeline where the problem was added so that the evaluator can replay the video segment if needed. Moreover, the tag area (Fig. 1h) allows the evaluator to create multiple tags and attach them to a problem description. VisTA stores all the created tags in a dropdown list so that the evaluator can reuse them.

5.3 Visualization of Problems and Features

VisTA visualizes ML-inferred problems on the *problem timeline* (Fig. 1c), following the idea of showing "condensed" information. This design hides the complexity of the raw verbalization and speech features that might be hard for the evaluator to understand in their analysis. As this is the primary augmented information to a think-aloud session video, it is placed directly under the video player to facilitate quick scanning. The long red vertical line on the problem timeline (Fig. 1e) indicates the current time of the video. Further, to allow the evaluator to access the raw features without being overwhelmed, VisTA only reveals the ML's main input features in a short time window (e.g., 10 seconds) around the current time in the video, instead of the entire video as in our initial design, on the *feature timeline* as shown in Fig. 1d and Fig. 3. The start and end of the window are marked with two short red vertical lines around the current time on the problem timeline (Fig. 1e). When the evaluator plays the video, the feature values on the feature timeline are dynamically updated. As this is a less demanded feature per pilot users' feedback, it is placed under the problem timeline.

When the evaluator pauses the video, VisTA shows a snapshot of the input features at the current time on the top of the problem timeline (Fig. 4b). It also highlights parts of the video that have the same features. For example, Fig. 4c,d,e contain the same features as the current time as shown in Fig.4b. To help the evaluator better assess how ML-inferred problems align with the highlight areas, we color-code

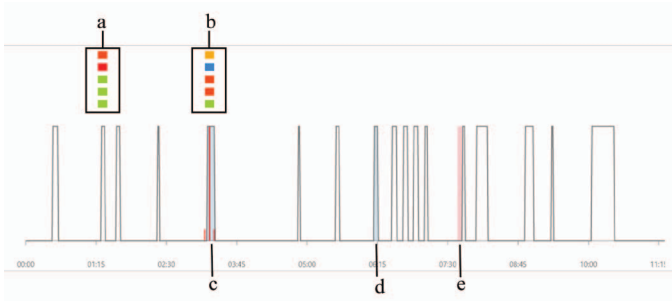

Fig. 4: The revised problem timeline. VisTA highlights all the segments that have the same set of features as the currently paused timestamp to help UX evaluators spot where else in the session the same features appear and how these areas align with ML-inferred problems.

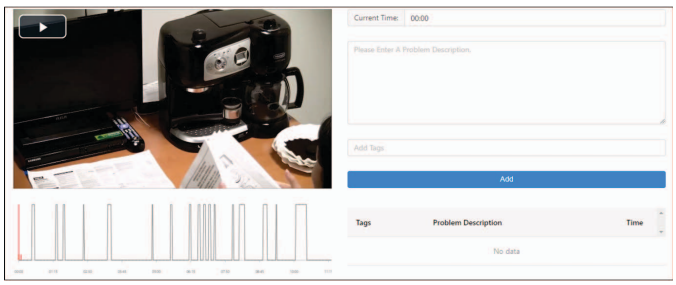

Fig. 5: VisTASimple has the same functions as VisTA except that it does not show the *feature timeline* or provide the filter function.

the areas where the ML detects problems in blue (Fig. 4c,d) and those where the ML detects no problems in pink (Fig. 4e). When the video is playing again, the highlight and the feature snapshot will disappear to avoid distraction. When the evaluator adds a problem, VisTA adds a feature snapshot on the top of the problem timeline (Fig. 4a) to help the evaluator remember the location of the problem and what the features look like at that time. Meanwhile, VisTA also adds an entry into the problem table (Fig. 1i). When the evaluator clicks on the snapshot, VisTA highlights all areas that have the same set of features.

Furthermore, as shown in Fig. 1j, VisTA also provides a filter function that allows evaluators to manually select a combination of features, which automatically highlights the areas on the problem timeline that have the same set of features. We posit that the highlighting areas would allow evaluators to better assess how the features of their choice align with the ML-inferred problems.

6 User Study

6.1 Design

To investigate how UX practitioners would use the problem timeline and the feature timeline in their analysis, we conducted a controlled laboratory study to compare different versions of VisTA. More specifically, we developed *VisTASimple* that only shows the problem timeline without the input features (Fig. 5), in order to better understand the effect of the feature timeline on a UX practitioner's analysis. This also helps us to investigate how it can affect the user interactions on the problem timeline. Moreover, to study the effect of the whole ML in the analysis process, we included a *Baseline* condition that shares the same user interface as VisTASimple except not having the problem timeline. Thus, UX practitioners do not have any access to ML.

Because there are potentially learning effects between conditions, we adopted a between-subjects design for the study. For example, after a participant used VisTA, she would know the ML's input features, which might prime her to consider these features in the other two conditions.

6.2 Participants

We recruited 30 UX practitioners from local UX communities at a large metropolitan area by posting advertisements on social media platforms. They participated in the study as *usability evaluators*. We randomly assigned them to the three conditions, thus each having 10 evaluators.

Table 2: The number of problems reported by evaluators (in $\mu(\sigma)$).

	Session 1	Session 2	Session 3
Baseline	3.9 (2.0)	6.2 (2.7)	13.8 (4.8)
VisTASimple	5.9 (2.8)	7.1 (2.4)	18.2 (6.6)
VisTA	6.7 (3.3)	8.4 (3.8)	21.2 (5.9)

They self-reported having 1-9 years of experience. The averages for the Baseline, VisTASimple, and VisTA conditions were the same: 3 years ($\sigma = 2, 3, 2$ respectively). Mann Whitney U test found no significant difference in the years of experience between conditions.

6.3 Procedure

We conducted the studies in a quiet office room with a 27-inch monitor connected to a laptop computer. After getting the evaluators' informed consent, the moderator explained that their task was to review three recorded think-aloud sessions to identify when users were confused, frustrated, or experienced problems. The three videos were about users operating on three different products (i.e, one website, one universal remote, and one coffee machine), and were randomly chosen from the dataset described in Sec. 4, and the same set of videos was used for all participants in the whole study.

In the beginning of the study, evaluators were demonstrated how to use the tool (Baseline, VisTASimple, or VisTA) by loading a trial think-aloud session, and the moderator answered any questions that they had. In each session, before evaluators analyzing the video, the moderator introduced the product and the task that the user worked on in the recorded video. The study lasted about 1.5 hours. We set the time for reviewing each video to be no more than three times of its playback length to ensure all the tasks would be completed within the study time. After each session, the moderator conducted a brief interview by asking how they analyzed the video. In the end of the whole study, evaluators filled in a questionnaire to rate their experience in using ML (for VisTA and VisTASimple) and their confidence in the problems that they identified on a 7-point Likert scale. Then, the moderator interviewed evaluators to further understand their confidence in the analysis results and their usages to the problem and the feature timelines (where appropriate). All interviews were audio-recorded. Each evaluator was compensated with $30.

6.4 Data Capture and Analysis

The software tool in all three conditions (i.e., Baseline, VisTASimple, and VisTA) recorded evaluators' interactions during the study. Specifically, it saved all the problem descriptions and their corresponding timestamps. We analyzed the reported problems to understand how evaluators performed in each condition. The tool also continuously recorded pairs of timestamps per second, (`SessionTime`, `VideoTime`), when evaluators were analyzing the sessions. This reflects the relationship between the timestamps in the analyzed video and in the study session. We analyzed this information to understand how evaluators reviewed the sessions (i.e., play, pause, rewind). We analyzed the evaluators' answers to the questions in the questionnaire to understand their usage of the tool. In addition, all interviews with the evaluators were recorded and transcribed. Two researchers coded the transcripts independently and then discussed to consolidate their codes. They then performed affinity diagramming to group the codes and identify the core themes emerged from the data.

7 Results

We present quantitative and qualitative results based on RQs in Sec. 3.

7.1 RQ1: How Would UX Practitioners Leverage ML in Their Analysis?

7.1.1 Number of reported problems

We counted the number of problems reported in each condition for each session (see Table 2). Evaluators found the highest number of problems in each session when using VisTA, followed by VisTASimple and then Baseline. One-way ANOVA found no significance in the number of problems identified between conditions for the first ($F_{2,27} =$

$2.70, p = .09, n_p^2 = .17$), and the second session ($F_{2,27} = 1.33, p = .28, n_p^2 = .09$), but found a significant difference for the last session ($F_{2,27} = 4.13, p = .03, n_p^2 = .23$). Post-hoc Bonferroni-Dunn test found significant difference between Baseline and VisTA.

7.1.2 How did evaluators use the *problem timeline*?

The interview data revealed four main ways of using the problem timeline. First, they used it as *an overview* to get a sense of the amount of potential problems and their distribution over the session even before playing the session. This overview information was useful for evaluators to get mentally prepared: *"Before the video starts, I looked at the chart to give me a heads up."-P39.* In the case of the third video where ML identified 17 problems, evaluators used this information to look out for *"big, overarching issues, instead of small little things."-P24.*

Second, evaluators used the problem timeline as *guides, reminders, and anticipations*. It was common that they may zone out while watching or listening to a long recorded test session, especially when hearing a long period of verbalizations of procedures that do not reveal any problem. In contrast, with the problem timeline, the "spikes" acted as reminders to pull them back and alert them to get ready. *"I'm using the spikes as anticipation...of when I should pay more attention."-P12. "I'll be like...a problem's coming up and then I'd pay attention and I will be waiting for the problem to pop up."-P26.*

Third, evaluators used the "spikes" on the problem timeline as *anchors* to facilitate their re-visitation. *"Then in the second [pass], I wanted to see all the ones that the machine learning highlighted [to] find things that I didn't notice on my first pass...I just would click where it starts going up, and then go through each one."-P21.* They also used it for grabbing representative quotes from users *"If I need to grab a quote, I will fast-forward to that part [the "spikes"]."-P12.*

Fourth, evaluators used the "spikes" to help them allocate attention. Some reported that they paid attention to all areas of the sessions but paid extra attention to the "spikes". Alternatively, because the "spikes" were visually salient, some paid more attention to the non-spike areas in their first pass of reviewing the sessions to catch any problems that ML might have missed. *"I should pay attention...when there's a long flat line...maybe they didn't pick up something. So I was listening to that part as well."-P20.*

7.1.3 How did evaluators use the *feature timeline*?

Evaluators in the VisTA condition had access to the *feature timeline* that visualizes the ML's main input features. But they usually allocated less attention to the feature timeline than the problem timeline. Given the short study time and the amount of videos to review, the filtering and highlighting functions (Section 5.3) were hardly used as evaluators mainly focused on leveraging the problem timeline and the feature timeline while reviewing the videos and writing problem descriptions. Evaluators mentioned that there was a learning curve to digest and leverage all the features and thus typically only considered the feature timeline in the second or third video session, when they became relatively familiar with the interface.

Evaluators felt that knowing the input features was helpful because this information allowed them to know what features were omitted by ML. Also, it allowed for them to better understand where ML could have missed problems, if the cues for a problem were primarily from the features that ML did not consider, such as visual cues. As a result, they could pay more attention to these features, which in turn allowed for better leverage of ML in their analysis.

In addition to employing the feature timeline to help better understand ML, some evaluators used the features directly in their own analysis. Among the features, categories were used more frequently as some observed that *"observation...could be a potential problem,"-P13* but *"reading [is] probably not so much of an issue."-P17.* On the contrary, evaluators had different opinions about pitch. Some thought it was helpful; for example, high pitch could reflect that the user was confused and raising a question. But others thought it was not a reliable signal without understanding the user's normal speaking behaviour. For example, some people tend to raise their tones toward the end of a sentence even if it is not a question.

In contrast, evaluators in the VisTASimple condition, who did not have access to the feature timeline, were asked if they had developed some understanding of the features that ML might have picked up. While many did not have any idea, some pointed out that ML might have used keywords or visual cues (e.g., how much movement the user had). These guesses were either only partially correct or not correct at all, which could prevent them from using the strategies that evaluators in the VisTA condition used.

Table 3: The number of times for pauses and rewinds (in $\mu(\sigma)$).

		Session 1	Session 2	Session 3
Pauses	**Baseline**	5.0 (3.3)	5.4 (3.5)	7.6 (6.9)
	VisTASimple	8.4 (5.5)	8.0 (4.7)	17.1 (6.4)
	VisTA	8.7 (7.3)	12.3 (7.6)	17.5 (7.5)
Rewinds	**Baseline**	4.5 (3.8)	4.3 (3.2)	7.6 (5.8)
	VisTASimple	20 (15.3)	15.3 (12.3)	18.0 (9.4)
	VisTA	5.2 (6.2)	9.0 (6.7)	9.8 (6.8)

Table 4: The evaluators' session review strategies based on passes.

	Session 1		Session 2		Session 3	
	1-Pass	2-Pass	1-Pass	2-Pass	1-Pass	2-Pass
Baseline	6	4	8	2	10	0
VisTASimple	4	6	6	4	8	2
VisTA	7	3	9	1	9	1

7.2 RQ2: How Would ML Influence UX Practitioners' Session Review Strategies?

7.2.1 Numbers of pauses and rewinds

We counted the number of times that evaluators paused and rewound the video in each session under each condition (see Table 3). Evaluators paused the most when using VisTA, followed by VisTASimple and then Baseline. One-way ANOVA showed that the difference was not significant for the first sessions ($F_{2,27} = 1.9, p = .17, n_p^2 = .12$), but was significant for the second ($F_{2,27} = 4.6, p = .02, n_p^2 = 0.25$) and the third session ($F_{2,27} = 6.4, p = .006, n_p^2 = .32$).

Further, evaluators rewound the most when using VisTASimple, followed by VisTA, and then Baseline. One-way ANOVA indicated that there was a significant difference for the first ($F_{2,27} = 7.7, p = .002, n_p^2 = .36$), the second ($F_{2,27} = 3.4, p = .049, n_p^2 = 0.20$), and the third session ($F_{2,27} = 5.0, p = .014, n_p^2 = .27$). The differences between VisTASimple and Baseline were significant, but the differences in all other condition pairs were not significant.

7.2.2 Session review strategies

We analyzed the pairs of timestamps (`SessionTime, VideoTime`) to further understand their session reviewing behaviour. We categorized typical behaviours by both *the number of passes on a video* and *the playback behaviours when going through a pass* (Fig. 6). In general, evaluators adopted one of the *one-pass* and *two-pass* approaches.

For the one-pass approach, there were three typical behaviours, namely *No-Pause-Write*, *Pause-Write*, and *Micro-Playback-Write*. No-Pause-Write means that evaluators kept the video playing while entering the problems identified (Fig. 6a). This behaviour was more common in the third video potentially due to the video length and the number of problems presented. For Pause-Write, evaluators paused the video while they enter the problems identified (Fig. 6b). With Micro-Playback-Write, evaluators repeatedly rewound and played a small section of the video while entering the problems identified (Fig. 6c). Evaluators who used VisTA or VisTASimple tended to adopt the Micro-Playback-Write strategy more than the Baseline. In particular, this strategy was adopted 6 times in Baseline, 18 times in VisTASimple, and 11 times in VisTA across all the sessions. It suggests that seeing the problem timeline had made them more cautious in their analysis. In addition, the Micro-Playback-Write strategy was adopted

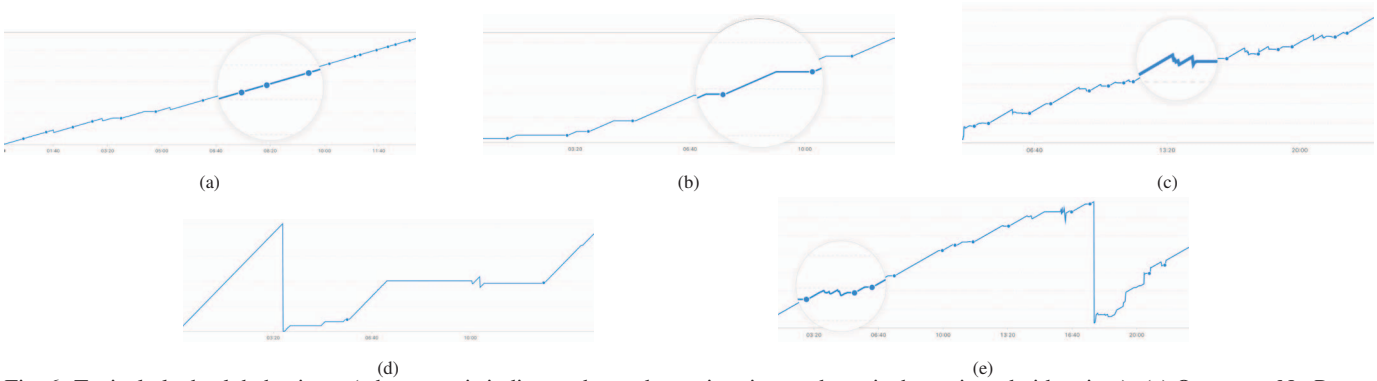

Fig. 6: Typical playback behaviours (where x-axis indicates the study session time and y-axis the reviewed video time): (a) One-pass: No-Pause-Write; (b) One-pass: Pause-Write; (c) One-pass: Micro-Playback-Write; (d) Two-pass: Overview-then-Write; (e) Two-pass: Write-then-Check.

more in VisTASimple than VisTA, suggesting that knowing the input features of ML allowed them to trust ML more and thus needed to rewind less frequently.

When evaluators adopted the two-pass approach, some used the first pass to gain an understanding of the context and to get a heads up of where the problems might be, i.e., *Overview-then-Write*. They sometimes played through the video without pausing or rewinding in the first pass if gaining context was the goal (Fig. 6d). On the other hand, some evaluators identified problems during the first pass and used the second pass as a chance to pick up the problems they might have missed or re-assessed issues they were not sure of, i.e., *Write-then-Check* (Fig. 6e). Table 4 shows the number of evaluators who used one-pass or two-pass approach. In all conditions, evaluators adopted one-pass and two-pass approaches and the proportions of the two were similar between conditions. The two-pass behaviour was more common in the first session than the last two, which may be due to the length of the videos, as the last video was the longest among all.

7.3 RQ3: How Would UX Practitioners Perceive and Manage the Relationship with ML?

7.3.1 Questionnaire responses

First, evaluators strongly agreed that they compared the ML-inferred problems in their analysis when using VisTASimple ($Mo = 7, Md = 7$) and VisTA ($Mo = 7, Md = 6.5$). They felt positively that they knew how to make use of the problem timeline when using VisTASimple ($Mo = 5, Md = 6$) and VisTA ($Mo = 6, Md = 6$). In general, evaluators agreed that ML helped them notice parts of the videos that they might have otherwise skipped if analyzing the videos without it when using VisTASimple ($Mo = 5, Md = 5$) and VisTA ($Mo = 5, Md = 5$). Also, evaluators would like to use VisTASimple ($Mo = 6, Md = 6$) and VisTA ($Mo = 5, Md = 6$) in future analysis.

Second, evaluators agreed more on the problems that the ML inferred ($Mo = 5, Md = 5$) than the problem-free areas that the ML inferred ($Mo = 3, Md = 3$) when using VisTA, and the difference was significance ($z' = -1.98, p' = .047$). In contrast, the difference was not different in VisTASimple.

Third, evaluators were confident that others would agree on the problems they identified: Baseline ($Mo = 6, Md = 5.5$), VisTASimple ($Mo = 6, Md = 6$), and VisTA ($Mo = 5, Md = 5$). Kruskal-Wallis test found no significant difference ($H' = 1.79, p' = 0.40$). They were also confident about the areas that they identified as problem-free: Baseline ($Mo = 4, Md = 5$), VisTASimple ($Mo = 5, Md = 5$), and VisTA ($Mo = 6, Md = 6$). No significant difference was found between conditions ($H' = 2.85, p' = 0.24$).

7.3.2 What were evaluators' attitudes toward ML?

Evaluators developed different perspectives on ML from their experiences. Four evaluators considered ML as a *colleague* or *coworker*, who could provide a *second perspective* on the identified problems. *"It might be picking up something that I had not been thinking about in a different sense...Could it be revealing something else I'm not picking up? Because I have my own confirmation bias."-P33.*

Two evaluators treated ML as a *backup* when ML agreed with them, which increased their confidence in the problems they identified. *"ML will back up my judgment, helped me confirm that there is a problem."-P17.* Three evaluators saw ML as an *aid* that helped them identify problems faster, not necessarily providing a different perspective that prompted them to reassess their disagreements. *"Use it for anticipation. When there is a prediction, I picked out problem faster. I don't consider it a different perspective."-P13.*

Additionally, four evaluators considered that there was a competition between them and ML. Evaluators had this feeling that they wanted to prove that they can do a better job and they had skills that ML may not necessarily possess. *"I didn't feel like it was smarter than me."-P36, "I want to feel I have skills too."-P17.*

On the other hand, three evaluators expressed concerns that using ML might cause them to be overly relying on it and get lazy in their analysis. *"If you don't care about your job you will just follow the chart...Someone still has to watch it (the video)."-P24.*

7.3.3 Amount of agreement and disagreement

Two researchers went through each problem that evaluators reported and compared its description and timestamps with the ML-inferred problems to determine if evaluators and the ML referred to the same problem. The agreement and disagreement of the reported problems between evaluators and the ML are shown in Table 5. One-way ANOVA found no significant difference in the number of problems that evaluators and the ML agreed for the first session ($F_{2,27} = 1.6, p = .22, n_p^2 = .1$) and the second session ($F_{2,27} = .9, p = .42, n_p^2 = .06$), but found significant difference for the third session ($F_{2,27} = 5.8, p = .008, n_p^2 = .30$). Post-hoc Bonferroni-Dunn test found significant difference between Baseline and VisTA. In contrast, there were no significant difference in the number of problems that were reported only by evaluators for the first ($F_{2,27} = .07, p = .93, n_p^2 = .005$), the second ($F_{2,27} = .006, p = .99, n_p^2 = .0005$), or the last session ($F_{2,27} = 2.2, p = .13, n_p^2 = .14$). Similarly, there were no significant difference in the number of problems that were reported only by the ML for the first ($F_{2,27} = 2.3, p = .12, n_p^2 = .15$), the second ($F_{2,27} = .66, p = .52, n_p^2 = .05$), or the last session ($F_{2,27} = 1.6, p = .22, n_p^2 = .11$).

7.3.4 How did evaluators deal with (dis)agreement with ML?

Evaluators felt that the agreement with the ML acted as confirmation and reassured them that they were correct with their reported problems. *"If I find a problem and the model also finds it, I feel more confident."-P26.* Evaluators also felt that seeing the agreement would make them *"pick up the problems faster."-P13.*

Evaluators generally understood that it was possible that ML is imperfect, *"Computer is not perfect...I don't expect it to be,"-P17*, and that ML can pick up different problems than they would. When it came to the disagreement, they considered false positives and false negatives of ML differently. When ML suggested a problem, they generally gave it a second thought even if it might be a false positive. *"I often wonder if I missed any problems, so it is safe to assume there is one [if the*

Table 5: The agreement and disagreement of reported problems between evaluators and the ML. ☺🤖: problems that both evaluators and the ML reported; ☺: problems that only evaluators reported; 🤖: problems that only the ML reported. Results are shown as $\mu(\sigma)$.

	Session 1			Session 2			Session 3		
	☺🤖	☺	🤖	☺🤖	☺	🤖	☺🤖	☺	🤖
Baseline	2.0 (1.1)	2.2 (2.1)	3.8 (1.2)	3.4 (.9)	3.4 (2.3)	.9 (.8)	9.3 (2.7)	4.4 (2.5)	8.4 (2.3)
VisTASimple	3.7 (1.7)	1.9 (1.5)	2.7 (1.5)	3.8 (2.0)	3.3 (2.1)	.8 (.8)	12.6 (4.2)	5.6 (3.8)	6.1 (3.1)
VisTA	4.0 (3.0)	2.1 (1.1)	2.3 (2.2)	5.1 (2.4)	3.3 (2.0)	.4 (.8)	15.1 (4.7)	6.8 (3.5)	4.0 (3.8)

ML detects it]."-P21. "It is not a big deal if the ML says there is a problem, I examine it and see nothing there."-P39. In contrast, if they thought that the think-aloud user encountered a problem but ML did not point it out, they generally considered that ML missed the problem and would more likely choose to trust themselves. *"By the third session, I started to really believe that the machine was just purely picking up more of the audio than the visual. So I think that's why...I gained a little bit more confidence."-P26.* In addition, they generally valued recall over precision, which is consistent with the findings of recent research (e.g, [26]). This can be explained by the fact that the goal for evaluators is to find potential problems, therefore it is safer to be overly-inclusive, which would introduce false positives, than missing potential problems, which would introduce false negatives.

It is also worth noting that evaluators in VisTASimple generally put less weight on the ML's predictions than those in VisTA when disagreements happened, which is probably because the ML in VisTASimple was more likely to be perceived as a "black-box." *"I don't know much about what it is based on and how developed the machine learning is, so I don't know how much I can trust it."-P18.*

7.3.5 What were the perceived limitations of the ML?

Evaluators pointed out a number of limitations based on their usage of ML. First, they noted that ML was often able to detect the moments when the user exhibited symptoms of a problem but did not pinpoint the start and end of the problem. However, observing the problem build-up process was important to fully understand it. *"There was...what I call...a lagging factor. I would have liked to see some of those issues highlighted earlier than some of these spikes on the timeline."-P14.*

Second, evaluators mentioned that ML did not understand the nuances in a user's personality. For example, some users may prefer to say negative words even when they did not experience too much of a problem. *"I don't think computer will pick up nuanced behaviours and personalities."-P17.*

Third, they felt that ML did not fully understand the context of users' actions. For example, ML had difficulty picking up repetitions in actions: when users did something repetitively, it could be a problem even though all the steps were performed correctly. Additionally, they felt that ML did not consider the number of steps taken to complete a task as a factor when detecting problems. For example, taking more steps than needed could mean a problem although the user completed the task successfully. Lastly, they felt ML did not fully comprehend the structures of the tasks (e.g., what (sub)tasks did users struggle with?).

8 Discussion

8.1 The Effect of ML on Evaluators' Analysis

Evaluators identified more problems when using VisTA than Baseline in all three sessions. The implication is that evaluators had more instances to understand potential problems when using VisTA. Such difference was significant for the third session, but not for the first two sessions (Sec. 7.1.1). One possible reason could be that as this was the first time evaluators had access to ML, they needed time to learn and understand how to leverage the ML-inferred problems in their analysis over the sessions. Evaluators mentioned that they either did not have much time to carefully consider the problem timeline or were still testing it in the first session. But over time, they were able to develop four general strategies (Sec. 7.1.2) to use the problem timeline. These strategies encouraged evaluators to be more cautious about their analysis, which was evident by the fact that evaluators using VisTA paused the videos significantly more than those using Baseline (Sec. 7.2.2). Another possible reason for non-significance in the first two sessions could be that these sessions were shorter and contained much fewer problems than the last one and thus the potential variations between conditions would also be smaller. Further research is needed to confirm whether the record think-aloud session's length influences ML's effectiveness.

Intuitively, an evaluator pointed out, *"Without ML, it is much easier to ignore and let go some issues."-P21.* When evaluators were watching a session to understand the development of a problem, a new problem might come up, which could take their attention away. If they did not rewind or pause the video in time, they could have missed the locations where they would otherwise want to follow up later. In contrast, the problem timeline acted as an overview, guides, anchors or anticipations, which facilitated evaluators with pinpointing the areas that they wanted to rewind and pause. This was more effective than using the Baseline to check the points that they might have missed. It is worth pointing out that the way in which evaluators used the problem and feature timelines is inherently tied to their session reviewing behaviour (e.g., pausing and rewinding), and is eventually tied to the number of reported problems. The significant difference in the numbers of problems and in the amounts of pausing and rewinding suggest that a ML-enhanced visualization is capable of helping evaluators become more cautious of their analysis and notice problems that they might have missed.

Despite the evaluators reported more problems when using VisTA than Baseline, they did not rewind the videos significantly more often. One potential explanation is that the evaluators' reported problems were also visualized at the corresponding timestamps on top of the problem timeline in VisTA (Fig. 4a). The visualization of the reported problems might also have acted as anchors, in addition to the ML-inferred problems on the problem timeline, that allowed the evaluators to better determine where they should rewind the video.

Although evaluators found more problems using VisTASimple than Baseline for all three sessions, One-way ANOVA did not find a significant difference. This could potentially suggest that having access to the feature timeline might play a role in encouraging evaluators to identify more problems. One reason could be that since evaluators in the VisTA condition knew what features were considered by ML, they could better infer when ML would make a mistake and focus on those features, such as visual cues, that ML did not consider. Another reason could be that evaluators leveraged the feature timeline as additional information in their analysis instead of merely using it to understand ML. However, individual differences between Baseline and VisTASimple could also come into play, as we used the between-subjects design and tested each condition with only 10 evaluators.

8.2 Attitudes toward ML

"Evaluator effect" refers to the fact that different evaluators might identify different sets of problems when analyzing the same session [23]. Although it is recommended to have more than one UX evaluator analyze a usability test session to reduce potential evaluator effect, fewer than 30% UX practitioners actually had an opportunity to work with others to analyze the same usability test session [15]. Our study reveals that a common attitude toward ML was to treat it as a "colleague" or a "coworker", who can provide a second perspective on their analysis or back up their identified problems. This finding points out an opportunity to leverage ML to help reduce the "evaluator effect" for UX practitioners, who often operate under resource and time constraints [7, 15]. Toward this goal, we have identified three factors to consider when designing a user interface that leverages ML to offer a second perspective to UX practitioners.

First, evaluators felt that knowing the severity of the problems that ML identified can help them to prioritize their analysis especially

when they are under time pressure to analyze a large amount of test sessions. Second, evaluators also felt that knowing the confidence level of ML in its inferences can be helpful. For example, they could filter out the low-confident inferred problems and focus more on the high-confident ones, especially when the session is long and has many inferred problems. Third, evaluators felt that ML would be more like a "colleague" if it could provide explanations for the detected problems. But what explanations are appropriate and how to generate them? Recent research suggested that the taxonomy for explaining ML to designers is likely *"to be radically different from ones used by data scientists"* [39]. In fact, evaluators who used VisTA felt that the current terms used for input features, such as category and sentiment, were too system-oriented and hard to interpret. They preferred the features to be expressed using layman terms, such as the level of surprise, excitement, or frustration. In addition, it might be beneficial to consider multiple explanations instead of seeking for the best one [38].

8.3 Reliance on ML

Three evaluators expressed the concern that this may make UX practitioners rely on ML too much, thus less diligent in their jobs. However, we did not find any evidence to support this. First, in all three conditions, evaluators identified problems that the ML did not identify, and there was no significant difference between conditions. Similarly, in all the conditions, evaluators disagreed on some of the ML-inferred problems and there was no significant difference between conditions either. These results suggest that evaluators did not just focus on the ML-inferred problems or took the words from the ML without scrutinizing them in the VisTA and VisTASimple conditions. Additionally, some evaluators even felt that there was a *competition* between them and the ML, making them subconsciously eager to prove that they could find more problems than the ML. It is, however, worth noting that as our study duration was short, no baseline trust with the ML had been established. Consequently, it is hard to determine whether evaluators would become over-reliance on ML or develop sustainable cooperative strategies in the long run.

Although none of the evaluators solely relied on the ML without putting in their own thought during analysis, we identified two ways in which evaluators wanted the ML to be presented. One way is to allow evaluators to analyze a test session by themselves in the first pass and then revealing the problem timeline to them in the second pass. In this way, the problem timeline would mainly help them confirm their judgment or double check if they might have missed any problem. The other way is to show the problem timeline all the time. The rationale for this design is that the two-pass reviewing process might not be practical especially when the session is long. This was evident that there were fewer evaluators who adopted the two-pass strategy for the third video, which was the longest among all (see Table 4). Although offering evaluators an option to turn on and off the problem timeline seems to be a compromised approach, it remains an open question how and when to best present ML to evaluators.

8.4 Trust in ML

We did not explicitly measure evaluators' trust in ML, however, we identified two factors from the interviews that could have affected their trust, including the *sophistication* and the *amount of disagreement*. The sophistication of ML is determined by the number of features that it considers (e.g., audio and visual features) and whether it understands the context of the task (e.g., the number of steps required to complete a task; meaningless repetitive user actions) or the personality of the user (e.g., the speaking behavior). Evaluators in all three conditions were fairly confident in their reported problems no matter how many problems they missed. This could suggest that UX practitioners might suffer from "confirmation bias" [33]. Confirmation bias can be mitigated by revealing the prior probability of events or input attributions [6, 29]. For example, it might be helpful to show evaluators the prior probability of catching all the problems from a test session (e.g., 70%) so that they might be more willing to reconsider their decisions when the ML disagrees with them. The goal of having ML's support is to encourage evaluators to scrutinize their analysis with the input of a different perspective from ML. It is, however, not to overly convince evaluators to agree with ML as it is still an open question whether increasingly agreeing with ML is beneficial for UX analysis. Another way could be to redesign the user interface to prompt evaluators to enter the features that they have considered and then ML could point out the features that they might have neglected. However, how to best design such systems that both deliver ML results and facilitate trust remains to be explored.

8.5 Future Work

We have identified four directions to explore in the future. First, evaluators felt that it would be informative to know the level of confusion or frustration (i.e., the severity of problems) and the confidence of ML for each identified problem, which could allow them to better prioritize their attention when time and resource is constrained. Future research is needed to design methods for detecting such information. Second, challenges need to be addressed for designing a visual analytics tool that effectively present all the information without overwhelming evaluators. One approach is to allow for turning on and off different functions. Our study results suggested some design considerations. For example, while some evaluators preferred to analyze think-aloud sessions without ML assistance in the first pass and only see ML-enhanced information in the second pass, others preferred to see ML-enhanced information in one pass to reduce time cost. Third, it is promising to explore ways of describing or explaining ML-inferred problems using a language familiar to UX practitioners. Future research should examine how UX practitioners communicate usability problems with their colleagues. Fourth, we used supervised learning to detect problems. Mixed-initiative interaction design, on the other hand, would allow a UX evaluator to annotate the ML's errors from which the ML can learn and evolve. Although promising, one potential caveat of learning from a UX evaluator is that the ML might behave more and more like the evaluator as the evaluator "corrects" the ML. Because one critical benefit of the ML is to offer a different perspective on the analysis, such an overly personalized ML, in the context of UX evaluation, would likely enhance the evaluator's confirmation bias instead of helping her spot potentially neglected areas.

9 Conclusion

We took the first step to explore how UX practitioners use, perceive and react to machine intelligence when analyzing recorded think-aloud sessions. We designed a visual analytics tool, VisTA, that presents ML-inferred problems and input features with timeline visualizations among other functions to facilitate UX practitioners with their analysis. Our three-session between-subjects controlled study compared VisTA, VisTASimple, and Baseline. Results showed that UX evaluators identified significantly more problems without needing to rewind the video more often when using VisTA than Baseline by the last session. Evaluators used the *problem timeline* as an overview, reminders, anticipations, and anchors to help them allocate their attention, spot areas that they might have otherwise neglected, and better revisit the videos. They used the *feature timeline* to understand what features were used and omitted by the ML and also used the features directly as an additional source of information in their analysis. Evaluators treated ML as a "colleague" who can offer a different perspective, as an aid that can make the analysis more efficient, or even as a "competitor" who encouraged them to spot more problems to "beat" it. In addition, evaluators both agreed and disagreed with ML-inferred problem encounters in all test conditions and did not seem to be overly reliant on ML. However, long-term deployment study is needed to validate this conjecture. Furthermore, they perceived the cost of false negatives of ML higher than that of false negatives and valued recall over precision. In addition, evaluators adopted three types of one-pass and two types of two-pass session reviewing strategies in each of the three conditions. Lastly, advanced features in VisTA, such as filtering and highlighting functions, were underused by evaluators. Therefore, in addition to detecting a richer set of information about problems (e.g., severity, explanation), it is also important to explore ways to deliver such information so that UX practitioners can better exploit it.

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *Proceedings of the Symposium on Operating Systems Design and Implementation*, pp. 265–283, 2016.

[2] T. Blascheck, M. John, K. Kurzhals, S. Koch, and T. Ertl. Va2: A visual analytics approach for evaluating visual analytics applications. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):61–70, 2016. doi: 10.1109/TVCG.2015.2467871

[3] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl. Visualization of eye tracking data: A taxonomy and survey. *Computer Graphics Forum*, 36(8):260–284, 2017. doi: 10.1111/cgf.13079

[4] P. Boersma. Praat: doing phonetics by computer. *http://www. praat. org/*, 2006.

[5] T. Boren and J. Ramey. Thinking aloud: Reconciling theory and practice. *IEEE transactions on professional communication*, 43(3):261–278, 2000.

[6] A. Bussone, S. Stumpf, and D. O'Sullivan. The role of explanations on trust and reliance in clinical decision support systems. In *Proceedings of the International Conference on Healthcare Informatics*, pp. 160–169. IEEE, 2015.

[7] P. K. Chilana, J. O. Wobbrock, and A. J. Ko. Understanding usability practices in complex domains. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2337–2346. ACM, 2010.

[8] L. Cooke. Assessing concurrent think-aloud protocol as a usability test method: A technical communication approach. *IEEE Transactions on Professional Communication*, 53(3):202–215, 2010.

[9] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang. Recovering reasoning processes from user interactions. *IEEE Computer Graphics and Applications*, 29(3):52–61, 2009. doi: 10.1109/MCG.2009.49

[10] G. Dove, K. Halskov, J. Forlizzi, and J. Zimmerman. Ux design innovation: Challenges for working with machine learning as a design material. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 278–288. ACM, 2017.

[11] M. Drouhard, N. Chen, J. Suh, R. Kocielnik, V. Pea-Araya, K. Cen, , and C. R. Aragon. Aeonium: Visual analytics to support collaborative qualitative coding. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 220–229, 2017. doi: 10.1109/PACIFICVIS.2017.8031598

[12] S. Elling, L. Lentz, and M. De Jong. Combining concurrent think-aloud protocols and eye-tracking observations: An analysis of verbalizations and silences. *IEEE Transactions on Professional Communication*, 55(3):206–220, 2012.

[13] K. A. Ericsson and H. A. Simon. *Protocol analysis: Verbal reports as data.* the MIT Press, 1984.

[14] M. Fan, J. Lin, C. Chung, and K. N. Truong. Concurrent think-aloud verbalizations and usability problems. *ACM Trans. Comput.-Hum. Interact.*, 26(5):28:1–28:35, 2019. doi: 10.1145/3325281

[15] A. Følstad, E. Law, and K. Hornbæk. Analysis in practical usability evaluation: a survey study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2127–2136. ACM, 2012.

[16] M. C. Fox, K. A. Ericsson, and R. Best. Do procedures for verbal reporting of thinking have to be reactive? a meta-analysis and recommendations for best reporting methods. *Psychological bulletin*, 137(2):316, 2011.

[17] M. Gillies, R. Fiebrink, A. Tanaka, J. Garcia, F. Bevilacqua, A. Heloir, F. Nunnari, W. Mackay, S. Amershi, B. Lee, et al. Human-centred machine learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 3558–3565. ACM, 2016.

[18] D. Gotz and M. X. Zhou. Characterizing users visual analytic activity for insight provenance. In *2008 IEEE Symposium on Visual Analytics Science and Technology*, pp. 123–130, 2008. doi: 10.1109/VAST.2008.4677365

[19] P. Hebron. *Machine Learning for Designer*. O'Reilly Media, Inc., 2016.

[20] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1189–1196, 2008. doi: 10.1109/TVCG.2008.137

[21] M. Hertzum, P. Borlund, and K. B. Kristoffersen. What do thinking-aloud participants say? a comparison of moderated and unmoderated usability sessions. *International Journal of Human-Computer Interaction*, 31(9):557–570, 2015.

[22] M. Hertzum, K. D. Hansen, and H. H. Andersen. Scrutinising usability evaluation: does thinking aloud affect behaviour and mental workload? *Behaviour & Information Technology*, 28(2):165–181, 2009.

[23] M. Hertzum and N. E. Jacobsen. The evaluator effect: A chilling fact about usability evaluation methods. *International Journal of Human-Computer Interaction*, 13(4):421–443, 2001.

[24] M. Hori, Y. Kihara, and T. Kato. Investigation of indirect oral operation method for think aloud usability testing. In *International Conference on Human Centered Design*, pp. 38–46. Springer, 2011.

[25] C. J. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*, 2014.

[26] R. Kocielnik, S. Amershi, and P. N. Bennett. 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*, CHI '19, pp. 411:1–411:14, 2019.

[27] C. Lewis. Using the'thinking-aloud'method in cognitive interface design. *Research Report RC9265, IBM TJ Watson Research Center*, 1982.

[28] J. R. Lewis. Usability: lessons learned and yet to be learned. *International Journal of Human-Computer Interaction*, 30(9):663–684, 2014.

[29] B. Y. Lim and A. K. Dey. Design of an intelligible mobile context-aware application. In *Proceedings of the 13th international conference on human computer interaction with mobile devices and services*, pp. 157–166. ACM, 2011.

[30] H. R. Lipford, F. Stukes, W. Dou, M. E. Hawkins, and R. Chang. Helping users recall their reasoning process. In *IEEE Symposium on Visual Analytics Science and Technology*, pp. 187–194, 2010. doi: 10.1109/VAST.2010.5653598

[31] S. McDonald, H. M. Edwards, and T. Zhao. Exploring think-alouds in usability testing: An international survey. *IEEE Transactions on Professional Communication*, 55(1):2–19, 2012.

[32] S. McDonald, T. Zhao, and H. M. Edwards. Dual verbal elicitation: the complementary use of concurrent and retrospective reporting within a usability test. *International Journal of Human-Computer Interaction*, 29(10):647–660, 2013.

[33] R. S. Nickerson. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of general psychology*, 2(2):175–220, 1998.

[34] J. Nielsen. *Usability engineering*. Elsevier, 1994.

[35] J. Nielsen. Thinking aloud: The# 1 usability tool. *Nielsen Norman Group*, 16, 2012.

[36] J. Nielson. Demonstrate Thinking Aloud by Showing Users a Video, 2014.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[38] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim. Designing theory-driven user-centric explainable ai. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pp. 601:1–601:15, 2019.

[39] Q. Yang. Machine learning as a ux design material: How can we imagine beyond automation, recommenders, and reminders? In *2018 AAAI Spring Symposium Series*, 2018.

[40] Q. Yang, N. Banovic, and J. Zimmerman. Mapping machine learning advances from hci research to reveal starting places for design innovation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 130. ACM, 2018.

[41] Q. Yang, A. Scuito, J. Zimmerman, J. Forlizzi, and A. Steinfeld. Investigating how experienced ux designers effectively work with machine learning. In *Proceedings of the 2018 Designing Interactive Systems Conference*, pp. 585–596. ACM, 2018.

[42] Q. Yang, J. Zimmerman, A. Steinfeld, and A. Tomasic. Planning adaptive mobile experiences when wireframing. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pp. 565–576. ACM, 2016.

[43] J. Zhao, M. Glueck, S. Breslav, F. Chevalier, and A. Khan. Annotation graphs: A graph-based visualization for meta-analysis of data based on user-authored annotations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):261–270, 2017. doi: 10.1109/TVCG.2016.2598543

[44] J. Zhao, M. Glueck, P. Isenberg, F. Chevalier, and A. Khan. Supporting handoff in asynchronous collaborative sensemaking using knowledge-transfer graphs. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):340–350, 2017. doi: 10.1109/TVCG.2017.2745279

[45] T. Zhao, S. McDonald, and H. M. Edwards. The impact of two different think-aloud instructions in a usability test: a case of just following orders? *Behaviour & Information Technology*, 33(2):163–183, 2014.