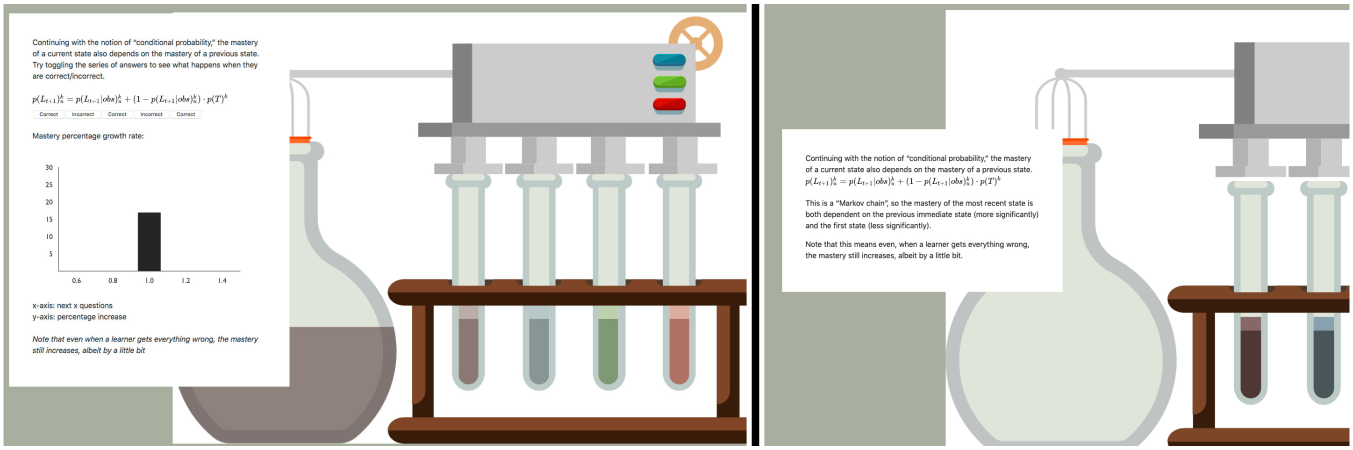
Assessing Post-hoc Explainability of the BKT Algorithm

Tongyu Zhou*
tz5@williams.edu
Williams College
Williamstown, MA, USA

Haoyu Sheng*
hs9@williams.edu
Williams College
Williamstown, MA, USA

Iris Howley
iris@cs.williams.edu
Williams College
Williamstown, MA, USA


Continuing with the notion of "conditional probability," the mastery of a current state also depends on the mastery of a previous state. Try toggling the series of answers to see what happens when they are correct/incorrect.

$p(L_{t+1})_u^k = p(L_{t+1}|obs)_u^k + (1 - p(L_{t+1}|obs)_u^k) \cdot p(T)^k$

Mastery percentage growth rate:

x-axis: next x questions
y-axis: percentage increase

Note that even when a learner gets everything wrong, the mastery still increases, albeit by a little bit

Continuing with the notion of "conditional probability," the mastery of a current state also depends on the mastery of a previous state.
$p(L_{t+1})_u^k = p(L_{t+1}|obs)_u^k + (1 - p(L_{t+1}|obs)_u^k) \cdot p(T)^k$

This is a "Markov chain", so the mastery of the most recent state is both dependent on the previous immediate state (more significantly) and the first state (less significantly).

Note that this means even, when a learner gets everything wrong, the mastery still increases, albeit by a little bit.

Figure 1: The interactive alchemy BKT explainable on the left, with a cropped view of the static explainable on the right.

ABSTRACT

As machine intelligence is increasingly incorporated into educational technologies, it becomes imperative for instructors and students to understand the potential flaws of the algorithms on which their systems rely. This paper describes the design and implementation of an interactive post-hoc explanation of the Bayesian Knowledge Tracing algorithm which is implemented in learning analytics systems used across the United States. After a user-centered design process to smooth out interaction design difficulties, we ran a controlled experiment to evaluate whether the interactive or static version of the explainable led to increased learning. Our results reveal that learning about an algorithm through an explainable depends on users' educational background. For other contexts, designers of post-hoc explainables must consider their users' educational background to best determine how to empower more informed decision-making with AI-enhanced systems.

This article was presented as a poster and appears in the Proceedings of the Third Annual ACM/AAAI Conference on Artificial Intelligence, Ethics, and Society.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; Visualization design and evaluation methods; • **Computing methodologies** → *Artificial intelligence.*

KEYWORDS

explainable AI, post-hoc explanations, interpretability of algorithms, communicating algorithmic systems, evaluation of xAI systems

ACM Reference Format:
Tongyu Zhou, Haoyu Sheng, and Iris Howley. 2020. Assessing Post-hoc Explainability of the BKT Algorithm. In *Proceedings of the 2020 AAAI/ACM Conference on AI, Ethics, and Society (AIES '20), February 7–8, 2020, New York, NY, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3375627.3375856

1 INTRODUCTION

In order for users to make informed, ethical decisions with the assistance of the algorithmic systems on which they rely, they must understand the algorithm's processes and therefore its potential flaws and biases [21]. The same is true with educational technology, which is increasingly being used in the classroom to make decisions, such as who is at risk of dropping out and on which topics to focus on that day [3]. This concern focusing on machine models' ability to be interpreted is seated within the machine learning literature as transparency and interpretability, and the approach to provide users with an understanding of their models is referred to as post-hoc explanations or explainable AI (xAI) [17].

In this paper, we investigate the development of two versions of a post-hoc explanation for Bayesian Knowledge Tracing (BKT), a complex algorithm used in universities across the United States.

[*] Both authors contributed equally to this research.

We used a qualitative user-centered design process to identify user needs and important factors to consider in building post-hoc explanations for complex algorithmic processes, and then quantitatively evaluated two versions of our explainable for learning.

2 RELATED WORK

Increasing numbers of systems leverage artificial intelligence (AI) algorithms into critical decision-making processes, such as criminal justice [2]. From within education, a growing movement toward student modeling, personalized learning environments, and learning analytics systems also leads to growing reliance on complex computational processes for decision-making in the classroom. Other research has uncovered that if users are not provided with an interpretable algorithm, they will invent their own "algorithmic imaginaries" to explain the model output they observe, regardless of its accuracy [5]. The increasing reliance on algorithms to assist in decisions has lead to increased interest in the fairness, accountability, and transparency of these algorithms from within the artificial intelligence community [15], [21], [11].

2.1 Explainables

Model interpretability is pointed to as a potential remedy for increasing transparency of complex computational models, but it is as yet unknown how to achieve the required level of interpretability to achieve the desired impact on user understanding and decision-making. Lipton (2018) introduces a desiderata of interpretability research, in which post-hoc interpretability is one property progressing toward the goals of trust, causality, transferability, and informativeness, as well as fair and ethical decision-making in machine learning models [21]. Of the four types of post-hoc interpretability detailed by Lipton (2018), explainables typically fall under the "Explanation by Example" category.

Based on this explanation by example category, the machine learning and visualization communities have begun creating what they call "explainables" to teach the concepts of particular algorithms. Explainables "that explain how AI techniques work using visualizations," to quote a recent workshop call for Visualization of AI[1], often take the form of interactive graphs or visualizations interspersed with paragraphs of explanatory text. However, this concept is not unique to machine learning nor the information visualization community, as very similar approaches can be found from within educational psychology under the name inquiry-based learning. In [12], inquiry-based learning is motivated as a requirement for understanding scientific inquiry and as a means to acquire, clarify, and apply an understanding of science concepts. The authors describe their collection of interactive learning technologies that provide interactive geosciences visualizations for novices to explore various atmospheric and meteorological sciences topics. These visualizations were integrated into a classroom curriculum and progressed from simple to complex activities and from specific instructions to open-ended tasks. Where a systemic application of inquiry-based learning often contains this scaffolding from the simple to the complex, explainables often mimic this process through a shorter process the community describes as going "up and down the ladder of abstraction" [2].

From within the machine learning community, discussion of how researchers might generate a rigorous science of interpretability focuses on methods with which to evaluate the interpretability of different models. This includes pointing out scientific understanding, safety, ethics, mismatch objectives, and multi-objective trade-offs as incompleteness in models that produce unquantified bias [11]. While the authors introduce a sample approach for understanding user expertise via the basic units of the explanation or cognitive chunks [22], they largely miss the modern literature on teaching and learning. Simply put, we can imagine the user as a learner and the post-hoc interpretation as the learning content, and this would allow xAI researchers to leverage the entire body of educational psychology and learning science research to achieve their goals of post-hoc explanations of complex algorithms.

While there is much research on measuring and evaluating what it means to know a concept from within the learning science field, this approach does not appear present in the work on explainables and algorithmic transparency via post-hoc explanation. A review of the example explainables listed in the IEEE VIS Workshop on Visualization for AI Explainability[1], for example, shows a series of explanatory text interspersed with interactive visualizations, but no accompanying evaluations in the research literature. In short, the explainables community does not yet appear to empirically evaluate whether their explainables successfully explain the concepts the designers had intended. And in many cases, it is not clear how the explainable designers identified the concepts necessary to explain.

2.2 Effects of Interpretability on Actions

Assuming the desired level of interpretability is achieved, it is also unknown how that interpretability impacts user behavior with algorithmically enhanced systems. Discussions within the ML community suggest that once a model increases its transparency, whether through post-hoc explanations or other techniques, this will lead to more realistic trust in ML systems and eventually to fairer decisions made with the assistance of ML models [11].

However, research in the learning sciences suggests that increased transparency in grading can lead to student dissatisfaction and distrust [17, 18]. When comparing three levels of system transparency in a high-stakes essay peer assessment context within an online learning environment, individuals who received their expected grade reported that their system trust was unaffected by the three transparency levels [18]. Individuals who received a lower grade on their essay than they expected reported reduced system trust, unless the grading algorithm was explained at the medium transparency level. High levels of system transparency, including a paragraph explaining how raw grading scores were algorithmically adjusted, yielded system trust indistinguishable from students experiencing the low transparency condition.

Other shifts in attitudes in fairness and trust are reported in additional self-report designs, as in [20]. The author used a social psychological self-report approach to measure perceived fairness,

[1] http://visxai.io/

[2] http://worrydream.com/LadderOfAbstraction

trust, and emotional response when a “decision-maker" is algorithmic or human [20]. Results suggest that algorithmic and human-made decisions were equally fair and trustworthy for mechanical tasks, but for human-based tasks, algorithmic decisions were perceived as less fair and trustworthy. This experiment was performed in an online environment using self-report measures, and while it is an informative first step towards understanding user perceptions towards algorithms that influence decision-making, next steps involve measuring how these perceptions influence user behavior.

This prior work shows that user trust, fairness, and positivity are measurably influenced by perceptions of algorithms, but it is unknown how well the users must understand the algorithm to achieve these impacts on perceptions. In this paper, we begin to investigate how explainable design can impact user understanding, using Bayesian Knowledge Tracing as our example algorithm.

2.3 Interactive Learning Experiences

Part of the intuition behind explainable design is that an interactive experience is more engaging than a static learning environment and allows for hypothesis testing [3]. This intuition is supported by education research literature, such as in Koedinger et al. (2015) which showed that interactive activities with feedback in an online course lead to students learning one standard deviation more than students using just informational assets like videos and text [19]. This work also aligns with a growing movement within education research showing that active learning activities increases student learning in science, engineering, and mathematics fields [14]. And while it is understood that interactive elements must be interspersed with passive learning elements, it is not yet confirmed for explainables that this is necessarily the case.

2.4 Bayesian Knowledge Tracing

BKT or Bayesian Knowledge Tracing was introduced in 1995 by Corbett & Anderson as a means to model students’ knowledge as a latent variable within technologically enhanced learning (TEL) environments [8]. The TEL maintains an estimate of the probability that the student has learned a particular set of skills based on their performance on problems, which is statistically equivalent to a 2-node dynamic Bayesian network. The development of BKT allowed for more accurate student modeling and more personalized learning opportunities for students. Decades of additional research on learner modeling followed, resulting in a variety of improvements to the BKT approach, including estimating individual parameters instead of skill-based parameters [26] and new ways to estimate the initial parameters [9] to much predictive success. BKT is used across the United States in TELs such as the Open Analytics Research Service [3], among others [16].

BKT predicts whether a student has mastered a skill, or not yet mastered it (either due to lack of data, or repeated failed attempts). This process requires a mapping from problems to skills. Mastery predictions require four parameters to calculate, with a probability of 0.95 typically being used as a cut-off to qualify as skill mastery:

1. **P(L_0)**: the probability the student already knew the skill
2. **P(T)**: the probability that the student learned after a learning opportunity
3. **P(G)**: the probability the student guessed correctly on an unknown skill
4. **P(S)**: the probability the student made a mistake and slipped on a known skill

One copy of each of the above parameters are used per skill. These parameters are usually fit through a variety of methods [9], are typically shared across an entire class of students, and are often not updated throughout the learning exercises. As a student proceeds through a lesson and answers problems correctly or incorrectly, BKT updates its estimates of predictions of mastery via the formulae below. First, the system sets the first probability to the initial probability that the student knew the skill a priori in Equation 1. Then, the conditional probability is computed using either Equation 2 or 3 depending on whether the student answered the problem correctly. This conditional probability is then used to update the probability of skill mastery as in Equation 4. BKT is a sufficiently complex algorithm as to not be easily understood, but it is also sufficiently approachable to explain as the parameters and their interaction are all known.

$$P(L_1) = P(L_0) \tag{1}$$

$$P(L_{n-1}|obs_n = corr) = \frac{P(L_{n-1}) * (1 - P(S))}{P(L_{n-1}) * (1 - P(S)) + (1 - P(L_{n-1})) * P(G))} \tag{2}$$

$$P(L_{n-1}|obs_n = incorr) = \frac{P(L_{n-1}) * P(S)}{P(L_{n-1}) * P(S) + (1 - P(L_{n-1})) * (1 - P(G))} \tag{3}$$

$$P(L_n|obs_n) = P(L_{n-1}|obs_n) + ((1 - P(L_{n-1}|obs_n)) * P(T)) \tag{4}$$

3 DESIGNING AN AI EXPLAINABLE

The main goals of our explainable were to render the artificial intelligence algorithm approachable to non computer scientists in the form of a playful experience. Instructors often do not have extra time to dedicate to system understanding, and so one of the main constraints on our prototype was that the designs be engaging and brief. We are working toward connecting one of these explainables to an existing BKT learning system, in order to provide algorithmic understanding to instructors and students of that system.

3.1 User-Centered Design Process for the Initial Explainable

The explainable design selected through a user-centered design process uses an alchemy metaphor to visualize the changes in parameters of the model and the predicted mastery levels. Figure 1 is a single screen from the final Interactive Alchemy explainable. In this tutorial, users interacted with buttons and watched the animations accompanying the descriptions. Vials representing the four parameters would fill a beaker representing the probability the student had mastered a skill. Much of the BKT algorithm was illustrated through the animations of the liquid rising and falling in the vials. This explainable was implemented in HTML and Javascript, resulting in approximately 20 unique screens.

This design was selected over a few other brainstormed ideas due to its in-depth explanations of additional concepts. In particular, predicted mastery was represented as a liquid in a beaker, leading to the liquid rising whether the hypothetical student answered

[3] https://blog.ncase.me/explorable-explanations/

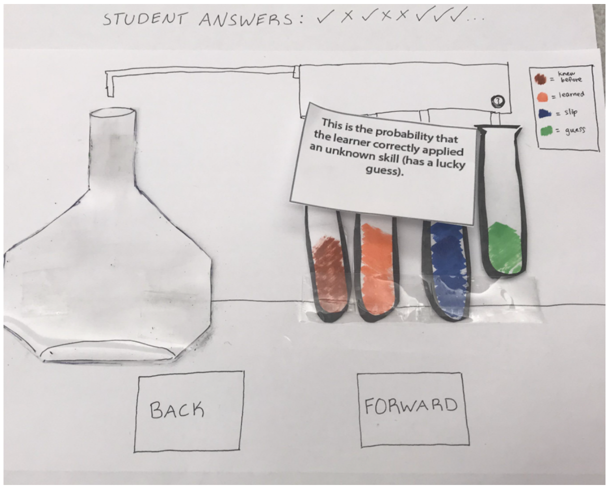

STUDENT ANSWERS: ✓ ✗ ✓ ✗ ✗ ✓ ✓ ✓ ...
This is the probability that the learner correctly applied an unknown skill (has a lucky guess).
?
= knew before
= learned
= slip
= guess
BACK
FORWARD

Figure 2: A low-fidelity prototype of the Alchemy explainable done with paper and markers.

a question correctly or incorrectly. This animation illustrates the BKT concept that every problem is practice in a skill, leading to an increased likelihood of mastery within that skill, regardless of the correctness of the student response. Our research team applied a user-centered design process to develop our initial interactive BKT explainable through the following steps:

Brainstorming & Sketching. Designs were initialized through brainstorming where the goal was to sketch at least nine ideas. Next, individual ideas were explained, refined, and edited in a group brainstorming process.

Low-fidelity Prototyping. Ideas from the brainstorming sessions were narrowed down to three and construction of low-fidelity paper prototypes began, as shown in Figure 2. After several iterations of prototyping within the research team, the paper prototypes were pilot-tested on fellow research assistants outside of the research team.

User Pilot Testing. Participants at this stage all had backgrounds in computer science at the undergraduate level, but none had experience with BKT or xAI. Each prototype was tested on two different users for each iteration following the process detailed in [23]. Users were instructed to think aloud and explain any difficulty in use or understanding that they encountered while interacting with the prototypes. They were also asked a series of knowledge check questions at the end of each session in order to ensure the prototypes were successful in teaching some BKT concepts.

Revisions & High-fidelity Prototypes. The prototypes were iteratively adjusted to address user feedback. After the third iteration, paper prototypes were converted into high-fidelity prototypes using click-through prototyping software.

Implementing the Designs. Two of the three designs were implemented at this stage, including the alchemy explainable. We then proceeded to run user tests with participants external to our department.

Participant Recruitment. With approval from our Institutional Review Board, we recruited eight participants from a small rural town in the northeastern United States via Craigslist, paper flyers, and the local college's online message board. With the consent of each participant, we audio recorded each 30-minute session.

Providing Task Context. After the first user study, participants were first presented with a brief lesson in a TEL, followed by an interactive quiz. The quiz provided immediate feedback on the participants' responses to questions about boxplots, and the final screen displayed a sample dashboard using BKT to predict the participants' mastery of the two boxplot skills. Participants were assured that their statistics skills were not evaluated, and were encouraged to guess if they found anything difficult, as the purpose was to demonstrate a context in which BKT is used so they could better understand the purpose of the explainables.

Usability Testing. We showed each participant two explainables in a thirty minute session, rotating through the six possible combinations to vary which explainables were shown, and which was shown first. A user test for one explainable typically lasted 10-15 minutes. While the feedback on the second shown explainable would be biased from the participants' exposure to the first explainable, we chose to leverage this opportunity to gather comparative feedback from users on multiple designs, similar to how prototype speed dating is leveraged in [10]. Once context for the BKT system was completed, participants were shown the first of two explainables and prompted to think aloud [24].

Semi-Structured Interview. After completing the explainable, we performed a semi-structured interview, asking for the participant's opinions on the format and content before a series of questions evaluating their knowledge retention. These knowledge retention questions were inspired by Bloom's Taxonomy [1] to ensure investigation into varying depths of understanding.

The above *Usability Testing* and *Semi-Structured Interview* processes were repeated a second time for a second explainable, asking for opinions and testing knowledge retention. Finally, participants were asked to compare the two explainables they saw, noting strengths, weaknesses, and preferences similar to [10].

Affinity Diagramming. When user testing was complete, a researcher listened to the audio recordings of the interviews and transcribed comments about the explainables resulting in approximately 90 of these raw comments. When all user comments were transcribed in this manner, an affinity diagramming process was followed as described in [4]. In user experience design contexts, the affinity diagram represents the scope of the user problem in performing a particular task and can be used throughout the user-centered design process to identify user needs in system designs.

We organized the quotes with a bottom-up approach by sorting them according to similar concepts. For example, we grouped notes that complained about an explainable having too many details together. We also gave each group a name that summarized the contents. A group with notes that expressed confusion surrounding the metaphor in an explainable was named "I don't understand the metaphor." We then sorted these groups by similarity into overarching categories. These categories form the major themes and issues users encountered while interacting with our explainables. Our final affinity diagram from this process is shown in Figure 3.

We identified three overarching categories through the affinity diagramming process which are shown alongside their associated themes in Table 1. The "Consider Individual Differences" section of our affinity diagram contained often-conflicting comments, which

Table 1: Affinity Diagramming Topics

Category	Themes
Consider Individual Differences	“The tutorial taught me something", “I don’t understand why we’re doing this", “I don’t understand the metaphor", “The metaphor worked", “The second tutorial confused me", “The second tutorial enlightened me", “I don’t know where my knowledge came from"
Refine the Level of Detail	“The details were helpful", “There was noticeable cognitive load", “Too many details make the tutorial overwhelming", “Too few details leave more questions"
Usability Design Principles	“It looks pretty and understandable", “The graphics were helpful", “The graphics were not helpful", “There are a few design changes I would make", “I expected change but didn’t get any", “Too many buttons makes it easy to mis-click", “Make the text readable", “Brief text is not intimidating", “Technical terms are intimidating"

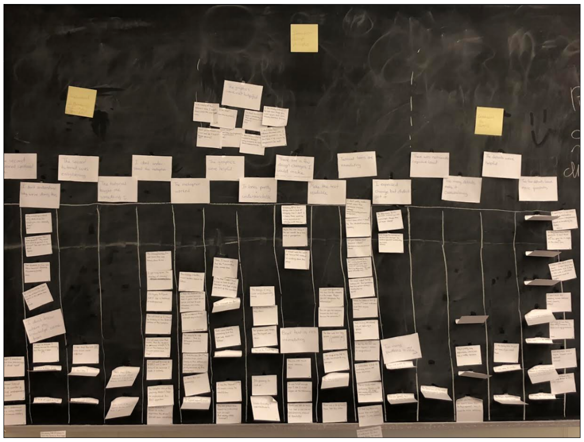


Figure 3: Photograph of the final affinity diagram with user comments grouped into themes via a bottom-up process.

we believe stem from each individual participant’s background and personal preferences. These comments covered topics such as comprehension of the tutorial materials. The “Refine the Level of Detail” category of our affinity diagram contained comments on satisfaction with the level of detail in our tutorials. “Usability Design Principles” contained comments about the appearance and function of our tutorials, which are typical details unearthed in usability tests.

3.2 Static and Interactive Explainables

After developing two explainables through this user-centered design process, our research team decided to focus on just one of the designs to better understand how different aspects of the design impacts user understanding of the algorithm. For this stage of the process, we selected the interactive alchemy design, and refined it according to user feedback from the previous phase as shown in Figure 1. We also implemented additional interaction features, such as allowing the user to see what happens to the algorithm output when a hypothetical students answers questions incorrectly or correctly, as well an opportunity to answer a multiple choice question and receive feedback.

We then produced an equivalent alchemy explainable with all of the interactive elements removed, which we call the *static alchemy explainable.* The interactive elements were replaced by a few sentences of explanatory text to ensure that users still covered the same concepts. Both of these new explainables were built using Javascript and Idyll, and proceeding through the content was maneuvered by scrolling down on the explainable. With a static and interactive version of our explainable, we could then examine whether an interactive explainable leads to increased learning over the static explainable, as the education research would suggest.

4 METHODS

We constructed a controlled experiment using Amazon Mechanical Turk. Participants were compensated $5.00, as we estimated the experiment to require 20-30 minutes of time. In total, we recruited 117 participants from North America, although 29 of these participants were dropped due to incomplete questionnaires, or answering one of two attention check questions incorrectly. 40 participants were randomly assigned to the interactive condition, and 48 to the static condition. 36% of participants self-identifed as female. 80% identified as white, 8% as black or African American, 6% as Asian, 5% as Latinx. 50% of participants had used an online platform to take a course previously.

4.1 Pretest

Participants first completed a pretest which consisted of introductory probability questions in order to assess quantitative familiarity. This section included questions such as “Which of the following cannot be a probability?” (with -0.002, and 1.01 being the correct answers), and other questions related to coin flipping and dice probabilities.

4.2 Posttest

After the pretest, participants were randomly shown one of the two alchemy explainables, and this was followed by a posttest. The posttest used Bloom’s Taxonomy as a guide for the complexity of the questions about BKT. Participants were asked basic comprehension questions, like the definition of p(init) and p(transit), but also more complex questions exploring the relationship between parameters such as “Sam makes spelling errors very frequently, rarely guesses at unknown words, and has reviewed the words very extensively beforehand. Given that three questions were correctly answered, it is very likely that Sam has mastered the vocabulary words.” These pre- and post-test questions were graded to produce a percentage score for each participant so learning could be measured. A sampling of the more complex questions from the posttest are included below:

(1) From the equations above, which one do you use to update the mastery when an observation is correct?
(2) Alex rarely makes arithmetic mistakes, rarely guesses, and has moderately reviewed algebra before. Given that three questions were correctly answered, how likely is it that Alex has mastered algebra?
(3) Sam makes spelling errors very frequently, rarely guesses at unknown words, and has reviewed the words very extensively beforehand. Given that three questions were correctly answered, it is very likely that Sam has mastered the vocabulary words.
(4) Morgan always guesses correctly, never makes mistakes, has an init probability of 0.6, and a transit probability of 1. What is the probability that Morgan has mastered the skill?
(5) Given everything else equal, if student A has a higher probability of guessing correctly than student B, whose mastery will grow more when they both answer a question correctly?
(6) Taylor rarely makes mistakes, rarely guesses, and reviewed the material very extensively beforehand. The student gets the first five questions wrong and the next four-in-a-row correct. BKT says that it is very unlikely for Taylor to have mastered the skill. Do you think this is a reasonable conclusion? Why/why not?
(7) What are some of the strengths of Bayesian Knowledge Tracing? What are some of its weaknesses?
(8) How fair or unfair is it for learners that the Bayesian Knowledge Tracing determines their skill mastery? Briefly explain the rating you give.
(9) How much do you trust that Bayesian Knowledge Tracing makes good-quality evaluations of a learner's learning process? Briefly explain the rating you give.

Partway through the above posttest comprehension items we include a two part attention check question adapted from [13]. The posttest was followed with a demographic post-questionnaire.

4.3 Data Analysis

Data was examined for normality, and we constructed a linear regression of pretest to posttest score for the entire population and calculated the residual for each participant. This residual score represents how much higher or lower the individual is relative to the regression line. In other words, it represents how much better or worse that participant performed than expected. This measure (or an Analysis of Covariance, controlling for pretest score) is preferable to computing learning by subtracting the pretest from the posttest, as the subtraction approach is prone to a ceiling effect. Someone who scores perfectly on the pretest cannot perform even better on the posttest, and a residual reflects this. For the purposes of this analysis "learning" is this pre-to-posttest residual score.

4.4 Results

We performed an Analysis of Variance statistical test, with condition as the independent variable and learning residual as the dependent variable. Initial analyses revealed no significant results of condition on learning. However, when Education Level is included as an interaction effect with condition, we find that the condition factor is significant, F(1, 9) = 4.71, p = 0.03, R^2 = 0.17 with mean learning residuals reported in Table 2. Table 2 shows high variability in learning between conditions based on education level. In particular, the two most extreme numbers are in the positive direction, and both for the static explainable condition. This is quite likely due to a small sample size.

Table 2: Learning by Condition and Education Level

n	Education	Condition	Learning	Standard Error
6	highschool	interactive	1.86	4.99
5	highschool	static	12.00	5.47
14	some college	interactive	-8.25	3.27
9	some college	static	-3.46	4.07
4	associate	interactive	-1.88	6.11
6	associate	static	3.95	4.99
11	bachelors	interactive	0.83	3.69
24	bachelors	static	0.66	2.50
5	masters	interactive	-1.70	5.47
4	masters	static	10.73	6.11

4.5 Discussion

In this experiment, our results ran counter to our hypotheses - the static explainable led to increased learning, but only when an interaction with education level was incorporated into the model. These results stress the importance of understanding the background of the user for which the explainable is designed. Different types of prior background may imply different approaches to designing an explainable.

The main limitation of this work is the relatively small sample size, relative to the 5 education levels. A larger sample or more targeted recruitment strategy is necessary for future work exploring user learning from AI explainables.

4.5.1 Future Work. There is a considerable amount of research on using interactive tutorials to teach content. However, work on explanations for particular artificial intelligence algorithms is quite new, and as such there are numerous exciting potential avenues for future work in this area. In particular, teaching a particular complex algorithm is so new, it is not yet known exactly how much understanding of an algorithm is necessary for impacting user decision-making. Cognitive Task Analysis is one possible method that can help systematically identify the knowledge components experts require to complete a task [7]. This will be informative for identifying the skills and concepts we want users to gain from interacting with an explainable. If an expert needs a particular concept for estimating whether the output of a machine learning model is flawed or biased, then that is a skill our explainables should teach.

Additional future work includes developing a BKT explainable with these principles incorporated. The first step involves performing expert Cognitive Task Analysis to identify the knowledge components that BKT experts use when making decisions with the assistance of BKT. A user-centered design process, similar to that described in this article, will be followed to adapt our explainables to target various levels of BKT understanding via a variety of flexible modules. Evaluation of the explainables will involve a pretest and posttest in order to investigate whether the users successfully learned the targeted concepts. Questionnaires will also be used to determine how the explainables change user self-reported trust and fairness perceptions of BKT. Final evaluation will examine how user decision-making in the classroom is impacted by an increased understanding of BKT. Further future work includes applying the above process to creating additional explainables for different algorithms, such as deep knowledge tracing [25].

5 CONCLUSION

In this article we presented results from our first investigation of designing explanations by example for BKT, a complex algorithm that predicts student knowledge within technologically enhanced learning environments. Through applying user-centered design practices and evaluating the system through a Mechanical Turk experiment, we discovered a considerable span of design considerations for xAI systems. By interpreting these results in light of learning science literature, we have produced the following main take-aways for designers of algorithmic explanations by example:

1. Different educational backgrounds require different approaches to designing an explainable.
2. Users had conflicting opinions of how much detail was an effective amount of detail to include in explainable tutorials. This points to users having varied prior knowledge and the importance of identifying a range of Zones of Proximal Development [6] that should be catered to pedagogically within the explainable.
3. Not all users were interested in deeply understanding the complex algorithm. It is important to provide an overview for all users, but also a details-on-demand option.
4. Users will be engaged differently with the explainables' content. Not all users find learning about algorithms intrinsically enjoyable. Motivating the explainable content from various perspectives can help, as can creating an engaging explainable experience.
5. Our two initial explainables targeted different depths of BKT understanding. However, it is unclear how much depth is necessary to achieve post-hoc 'interpretability.' Explainable designers should identify which algorithmic concepts are important to achieve their desired outcomes, and then teach those concepts.

This first user-centered design study and Mechanical Turk experiment has provided initial insight for designing instructive, interactive explainables with usability and knowledge-building at the center of the process. Going forward, we have plans to more systematically determine how much algorithmic understanding is necessary to achieve shifts in attitudes of system trust as well as decision-making behavior.

ACKNOWLEDGMENTS

Funding for this research was provided by the IIS:Cyber-Human Systems program: NSF CISE award #1849984.

REFERENCES

[1] Lorin W Anderson, David R Krathwohl, Peter W Airasian, Kathleen A Cruikshank, Richard E Mayer, Paul R Pintrich, James Raths, and Merlin C Wittrock. 2001. A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives, abridged edition. *White Plains, NY: Longman* (2001).

[2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. 2016. Machine bias: There's software used across the country to predict future criminals. *And it?s biased against blacks. ProPublica* (2016).

[3] Jonathan Bassen, Iris Howley, Ethan Fast, John Mitchell, and Candace Thille. 2018. OARS: exploring instructor analytics for online learning. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*. ACM, 55.

[4] Hugh Beyer and Karen Holtzblatt. 1997. *Contextual design: defining customer-centered systems.* Elsevier.

[5] Taina Bucher. 2017. The algorithmic imaginary: exploring the ordinary affects of Facebook algorithms. *Information, Communication & Society* 20, 1 (2017), 30–44.

[6] Seth Chaiklin. 2003. The zone of proximal development in Vygotsky?s analysis of learning and instruction. *Vygotsky?s educational theory in cultural context* 1 (2003), 39–64.

[7] Richard E Clark, D Feldon, Jeroen JG van Merriënboer, Kenneth Yates, and Sean Early. 2008. Cognitive task analysis. *Handbook of research on educational communications and technology* 3 (2008), 577–593.

[8] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.

[9] Ryan SJ d Baker, Albert T Corbett, and Vincent Aleven. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *Intelligent Tutoring Systems*. Springer, 406–415.

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

[11] Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning.(Feb. 2017). *arXiv preprint stat.ML/1702.08608* (2017).

[12] Daniel C Edelson, Douglas N Gordin, and Roy D Pea. 1999. Addressing the challenges of inquiry-based learning through technology and curriculum design. *Journal of the learning sciences* 8, 3-4 (1999), 391–450.

[13] Serge Egelman and Eyal Peer. 2015. Scaling the security wall: Developing a security behavior intentions scale (sebis). In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2873–2882.

[14] Scott Freeman, Sarah L Eddy, Miles McDonough, Michelle K Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences* 111, 23 (2014), 8410–8415.

[15] Wayne Holmes, Duygu Bektik, Denise Whitelock, and Beverly Park Woolf. 2018. Workshop on: Ethics in AIED: Who cares?. In *Proceedings of the Nineteenth (2018) Conference on Artificial Intelligence in Education*. http://oro.open.ac.uk/id/eprint/53443

[16] Kenneth Holstein, Bruce M McLaren, and Vincent Aleven. 2018. Student learning benefits of a mixed-reality teacher awareness tool in AI-enhanced classrooms. In *International Conference on Artificial Intelligence in Education*. Springer, 154–168.

[17] Kartik Hosanagar and Vivian Jair. 2018. We Need Transparency in Algorithms, But Too Much Can Backfire. *Harvard Business Review* (Jul 2018). https://hbr.org/2018/07/we-need-transparency-in-algorithms-but-too-much-can-backfire

[18] René F Kizilcec. 2016. How much information?: Effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2390–2395.

[19] Kenneth R Koedinger, Jihee Kim, Julianna Zhuxin Jia, Elizabeth A McLaughlin, and Norman L Bier. 2015. Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. In *Proceedings of the second (2015) ACM conference on learning@ scale*. ACM, 111–120.

[20] Min Kyung Lee. 2018. Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management. *Big Data & Society* 5, 1 (2018), 2053951718756684.

[21] Zachary C Lipton. 2018. The Mythos of Model Interpretability. *Queue* 16, 3 (2018), 30.

[22] Ian Neath. 1998. *Human memory: An introduction to research, data, and theory.* Thomson Brooks/Cole Publishing Co.

[23] Carolyn Snyder. 2003. *Paper prototyping: The fast and easy way to design and refine user interfaces.* Morgan Kaufmann.

[24] Maarten W Van Someren, Yvonne F Barnard, and Jacobijn AC Sandberg. 1994. *The think aloud method: a practical approach to modelling cognitive processes.* Londen: Academic Press.

[25] Lisa Wang, Angela Sy, Larry Liu, and Chris Piech. 2017. Deep knowledge tracing on programming exercises. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*. ACM, 201–204.

[26] Michael V Yudelson, Kenneth R Koedinger, and Geoffrey J Gordon. 2013. Individualized bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education*. Springer, 171–180.