How Do We Read Formal Claims? Eye-Tracking and the Cognition of Proofs about Algorithms

Hammad Ahmad*, Zachary Karas†, Kimberly Diaz‡, Amir Kamil§,
Jean-Baptiste Jeannin¶ and Westley Weimer‖
University of Michigan, Ann Arbor
*hammada@umich.edu, †zackar@umich.edu, ‡kkhalsa@umich.edu, §akamil@umich.edu,
¶jeannin@umich.edu, ‖weimerw@umich.edu

2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE) | 978-1-6654-5701-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSE48619.2023.00029

***Abstract*—Formal methods are used successfully in high-assurance software, but they require rigorous mathematical and logical training that practitioners often lack. As such, integrating formal methods into software has been associated with numerous challenges. While educators have placed emphasis on formalisms in undergraduate theory courses, such courses often struggle with poor student outcomes and satisfaction. In this paper, we present a controlled eye-tracking human study ($n = 34$) investigating the problem-solving strategies employed by students with different levels of incoming preparation (as assessed by theory coursework taken and pre-screening performance on a proof comprehension task), and how educators can better prepare low-outcome students for the rigorous logical reasoning that is a core part of formal methods in software engineering. Surprisingly, we find that incoming preparation is not a good predictor of student outcomes for formalism comprehension tasks, and that student self-reports are not accurate at identifying factors associated with high outcomes for such tasks. Instead, and importantly, we find that differences in outcomes can be attributed to performance for proofs by induction and recursive algorithms, and that better-performing students exhibit significantly more attention switching behaviors, a result that has several implications for pedagogy in terms of the design of teaching materials. Our results suggest the need for a substantial pedagogical intervention in core theory courses to better align student outcomes with the objectives of mastery and retaining the material, and thus bettering preparing students for high-assurance software engineering.**

***Index Terms*—formalism comprehension, student cognition, eye-tracking, facial behavior analysis, human study**

I. Introduction

Formal methods have long been used to provide rigorous guarantees for software engineering [1] (e.g., ensuring that executions of a program never reach an invalid state), and have been incorporated into various core stages of the software process. Successful applications of formal methods to software include requirements elicitation [2], software specification (e.g., *design by contract* [3]), software design (e.g., VDM [4]), software verification [5, 6], testing [7], and maintenance (e.g., legacy code at Microsoft makes use of assertions [8]). Unfortunately, many formal methods require advanced mathematical training and theorem proving skills that practitioners typically lack [9].

Given the extensive use of, and increased opportunities for, formal methods in software engineering [10], educators have been putting an increased focus on *formalisms* (e.g., proofs of algorithmic properties, runtime complexity analyses, etc.) in undergraduate computer science curricula to prepare future developers for logical algorithmic reasoning [11]. Unfortunately, despite the emphasis placed on formalisms in undergraduate computer science theory courses, students have historically struggled with course outcomes (e.g., in terms of final grades, mastery and retention of material, etc.). For instance, publicly-available survey data from a university in the US highlight a trend of dissatisfaction and low outcomes from core theory courses focusing on formalisms [12, 13]. Given the difficulty associated with having engineers integrate formal methods into the software process [14], it remains important for educators to ensure that future practitioners are trained in logical reasoning skills, especially as they relate to code.

We hypothesize that understanding how people less familiar with formalisms think about formal methods and proofs of algorithmic properties is critical to how educators should teach formalisms. For instance, since the most vulnerable population groups with non-traditional backgrounds are also most likely to drop the computer science major [12], educators need to make sure the needs of such groups are not overlooked. It also indirectly impacts how high-assurance software engineering firms might train new workers. One way to acquire this understanding is through the investigation of the *cognition* (e.g., problem-solving strategies, cognitive load, visual attention, etc.) for computer science students while performing formalism comprehension tasks.

Previous work has used methodologies like eye tracking, occasionally coupled with functional magnetic resonance imaging or functional near-infrared spectroscopy [15, 16], to investigate student cognition for computer science tasks relevant to software engineering, including reading [17] and writing code [18], manipulating data structures [19], and reviewing code [20]. Researchers have also examined the cognitive models associated with higher-level math tasks [21], including number processing and arithmetic. However, since formal methods fundamentally differ from other software engineering processes in their focus on mathematical and logical reasoning instead of coding, lessons learned from cognition for coding tasks may not clarify *how* students comprehend formalisms, *what* distinguishes an expert in formal reasoning from a novice, and *how* educators can better train students for formal reasoning for software.

We propose to use eye-tracking to gather insights into (i) the

1558-1225/23/$31.00 ©2023 IEEE
DOI 10.1109/ICSE48619.2023.00029

problem-solving strategies for such formalism comprehension tasks employed by students with different levels of familiarity with, or *incoming preparation* for, formal methods, and (ii) how educators can better prepare struggling students for the rigorous logical reasoning required by high-assurance software engineering. We consider both prior coursework and current performance in our definition of incoming preparation. We first ask participants to outline the number of computer science theory courses covering formalisms (e.g., derivations and proofs) they have either completed with passing grades or are currently taking. We then tested participants to identify the mistake in a proof taken from an undergraduate textbook [22]. We partition our sample based on whether a participant has taken more than the median number of courses in our sample *and* passed the screening test (see Section V-A for a more in-depth discussion on incoming preparation). Since eye-tracking methods allow for the possibility of adding a webcam to examine facial behavior as an additional information source in the context of formalism comprehension, we also employ *facial behavior analysis* to investigate behavioral differences between students with different outcomes.

We performed a controlled experiment investigating how students read and assess formal proofs about algorithms for correctness. We recruited 34 participants with varying levels of expertise with formalisms to perform these comprehension tasks. Participants were presented with pseudocode algorithms from a widely used undergraduate textbook [22], a theorem about the algorithm with an accompanying formal proof, and a graphical illustration of the algorithm or proof. Participants were asked to evaluate the presented proofs for correctness.

Contrary to conventional wisdom suggesting that students with greater incoming preparation achieve better outcomes for STEM courses [23, 24, 25, 26], we found no evidence that students with higher incoming preparation perform better at these formalism tasks ($p = 0.96$, Cohen's $d = 0.007$). We further find no evidence that student experience reports are accurate predictors of outcomes for formalism comprehension tasks ($\tau = 0.21, p = 0.15$), or that students are able to correctly identify the parts of a formalism presentation most pivotal to understanding a proof. Our results also indicate more-prepared students employ different problem solving strategies, with an increased visual attention on proof prose text ($p = 0.005$) and correct ($p = 0.03$) and distractor ($p = 0.038$) answer choices, but this ultimately does not affect task outcomes.

We do find, however, that higher outcome students demonstrate significantly more attention switching behaviors (i.e., frequently going back and forth between presented materials) ($p = 0.002$), and are more likely to perform better at proofs by induction ($p = 0.01$) and recursive algorithms ($p = 0.006$) compared to lower outcome students. Our results argue for the need for pedagogical intervention in theory courses to ensure student outcomes are aligned with the objectives of better teaching formal reasoning to undergraduates and preparing future software engineers for formal methods.

The main contributions of this paper are:

- A controlled experimental study investigating student cognition for computer science formalisms.
- Experimental evidence that suggests incoming preparation does not predict outcomes for formalism comprehension tasks, and that students with higher outcomes employ different problem-solving strategies and exhibit better performance for certain types of proofs and algorithms.
- Recommendations for educators to better prepare future software engineers for logical reasoning, including designing teaching materials to facilitate going back and forth between the presented content with ease, and emphasizing proofs by induction.
- An exploratory discussion on the use of objective facial behavior metrics in the context of pedagogy.
- A publicly-available dataset for future studies investigating cognition for computer science formalisms.

II. Background and Motivation

In this section, we summarize the techniques related to the eye-tracking and facial behavior analysis measures for a general software engineering audience, and provide a motivating example for our work.

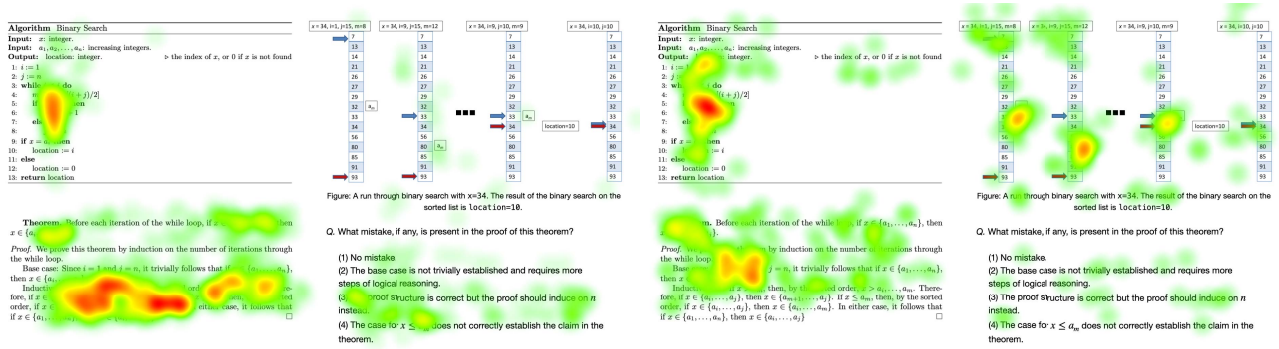
A. Eye-Tracking

Eye-trackers are non-invasive, cost-effective, and easy-to-use devices that measure visual attention and effort in variety of tasks, including human-computer interactions [27], software engineering [28, 29, 30], and marketing [31].

Modern eye-tracking cameras measure and track a participant's eyes and use event detection algorithms to report *gaze data* that is then analyzed with respect to pre-defined *areas of interest* (AOIs) in a stimulus. AOIs are typically manually defined by an experimenter based on the nature of the study [32, 33].

Two aspects of gaze data, based on ocular behavior, can clarify cognitive load and task difficulty. A *fixation* is an eye gaze that lasts for approximately 200–300ms on a specific AOI and results in the focus of visual attention on the AOI. The majority of information processing for humans occurs during fixations [34, 35] and a small number of fixations usually suffices for a human to process a complex visual input [32, 36]. As such, fixation data is widely used to measure cognitive load for different tasks, with longer fixations and higher number of fixations indicating higher cognitive load [29, 30]. A *saccade* is a rapid eye gaze movement (40–50ms) that occurs between fixations on AOIs, and often does not correspond to cognitive processing [34, 36]. The *regression rate* is the ratio of backward or regressive saccades (e.g., leftward in left-to-right text reading) to the total number of saccades, and higher regression rates often indicate increased difficulty in performing and completing a task [37].

Modern eye-trackers can also report the *pupil diameter* of participants. Pupil diameter has widely been used in the context of eye-tracking to approximate the cognitive load for participants working on study tasks [38, 39, 40], with higher pupil diameters indicating increased cognitive load [41].



(a) Higher incoming preparation participant heatmap

(b) Lower incoming preparation participant heatmap

Fig. 1: Visual eye-gaze heatmaps for a stimulus shown to two participants with different incoming preparations. The more-prepared participant focuses visual attention primarily on the proof and answer choices, while the less-prepared participant focuses on the algorithm and figure. Both participants choose the wrong answer.

B. Facial Behavior Analysis

Facial expressions constitute an important channel of non-verbal communication in humans [42], and facial behavior analysis has been increasingly used to facilitate human-computer interactions in an array of applications [43], including education [44, 45].

Facial behavior analysis often involves the detection, via cameras and image processing, of *facial Action Units* (AUs) [46] that correspond to individual components of facial muscle movement. The *Facial Action Coding System* (FACS) is an anatomically-based model used to describe any visually-discernible facial movement in terms of AUs [47].

While care must be taken to avoid linking facial behaviors directly to human emotions without considering context [48, 49], AUs hold the benefit of being objective measures of facial muscle activity, and can be used independently of any interpretation for higher-order modeling of facial expressions and behavior [50].

C. Motivating Example

We desire a deeper understanding of the problem-solving strategies for formalism comprehension tasks employed by students with different levels of incoming preparation, and how educators can better prepare struggling students for the use of formal methods in software engineering. Traditional metrics, such as evaluating student transcripts or self-reports, are not as effective at teasing apart the problem-solving strategies employed by higher performing students (e.g., [51]). We hypothesize that eye-tracking can serve as a cost-effective, insightful methodology to investigate the factors that result in better outcomes for formalism comprehension tasks. For instance, struggling students may demonstrate *higher regression rate* for (i.e., re-read text and figures more frequently), or *increased visual attention* to, certain aspects of a formalism presentation, suggesting greater difficulty completing the task.

As an indicative example, we present a snapshot of the strategies (in terms of visual attention) employed by two students with different incoming preparations for undergraduate theory courses (Figure 1). Figure 1a shows the *visual heatmap*, constructed from gaze data collected by an eye-tracker, for a participant with higher incoming preparation for theory courses. The heatmap indicates a significant proportion of visual attention to the proof text and answer choices (lower left and right quadrants respectively). By contrast, the visual heatmap for a participant with lower incoming preparation suggests a comparatively increased emphasis on the algorithmic pseudocode and figures (upper left and right quadrants respectively). While the increased focus by a less-prepared participant on the algorithm and figures aligns with instructor expectations, one would also expect a more-prepared participant focusing attention on the proof text to achieve better response accuracy. Quite surprisingly, we find that not only do both participants fail at correctly identifying the presence of mistakes in the proof, but also that this trend of no correlation between traditional measures of incoming preparation and task outcomes extends to the entirety of our participants (see Section V-A for a discussion on preparation and outcomes). Note that our use of heatmaps is intended to present a snapshot of what participants focus on, and is not the sole point of comparison between participants with different outcomes. Our results in Section V incorporate eye-tracking metrics better suited to understanding the complete picture.

Given that neither incoming preparation nor the strategies employed by more-prepared students is sufficient at teasing apart factors that result in better outcomes for such formalism comprehension tasks, we turn to eye-tracking to clarify factors correlated with student successes. Our results help us better understand what makes students succeed, and how educators can teach formalisms in a way that may potentially better prepare future engineers to reason about formal methods.

III. Experimental Methodology

Our experiment centers on a human study in which participants answered questions about computing formalisms (al-

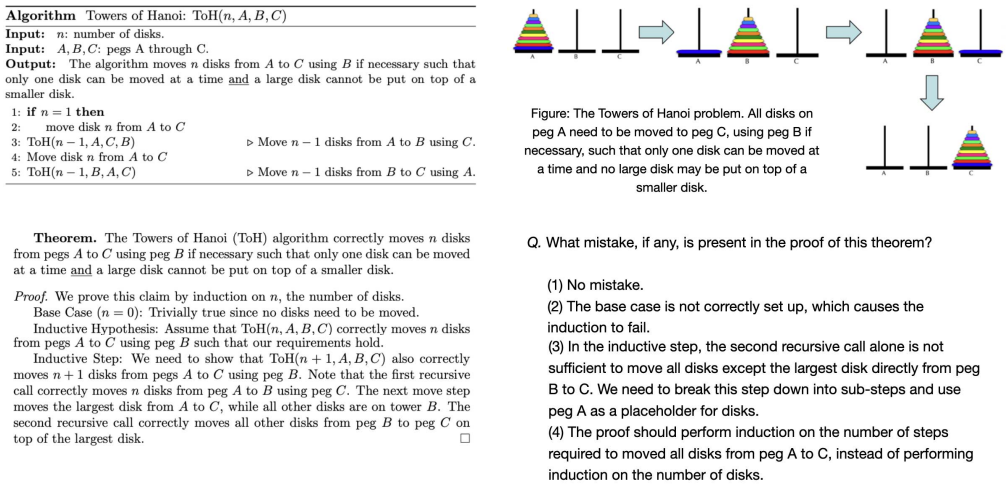


Fig. 2: A sample formalism comprehension stimulus for the Towers of Hanoi problem. The algorithm (upper-left), figure (upper-right), theorem and proof (lower-left) and correct and distractor answer choices (lower-right) represent the six AOIs for the stimulus. The correct answer is (2): the base case should apply when $n = 1$, and more reasoning is required to establish the claim for the base case.

gorithms, theorems, proofs, and figures) while subjected to eye-tracking and facial behavior measurement. We make the replication materials for our study (including the pre- and post-questionnaires, stimuli, and de-identified raw data) publicly available at https://doi.org/10.5281/zenodo.7626901.

A. Participant Recruitment

We recruited 34 undergraduate and graduate computer science students at the University of Michigan in an IRB-exempt study. Of our 34 participants, 23 identified as men, while 11 identified as women. Breaking down our participants by class standing, we recruited 5 first-year students, 10 sophomores, 12 juniors, 6 seniors, and one graduate student. We required participants be over 18, have completed an undergraduate discrete mathematics course, and be either enrolled in or have completed an undergraduate data structures and algorithms course. In addition, to reduce noise in the recorded gaze data, we encouraged (but did not require) our participants to wear contact lenses in lieu of glasses to the experiment session where possible. Participants were compensated $25.

B. Materials and Design

Participants were asked to complete a sequence of formalism comprehension tasks. Each individual task stimulus consisted of an algorithmic solution to a problem commonly taught in core computer science undergraduate courses, a theorem for that algorithm, an accompanying proof of that theorem, and a relevant graphical illustration (or figure). Each formalism comprehension task presented four multiple-choice answers for the presence of mistakes in the proof, of which only one was correct and three were distractors.

We seeded each proof with mistakes commonly made by undergraduate students in discrete mathematics courses (e.g., incorrect base case for proof by induction, logical contradictions in deductive reasoning, arithmetic errors leading to incorrect conclusions, etc.), and asked participants to evaluate each proof for correctness. For each algorithm, participants were always given the option to indicate that a presented proof contains no mistakes.

For our study, we presented 7 algorithms taken from a commonly-used undergraduate discrete mathematics textbook [22]: binary search, greedy change-making, merge-sort, Towers of Hanoi, greedy job scheduling, in-order tree walk, and the Halting Problem. Each algorithm was accompanied by a theorem and a proof copied verbatim from the textbook prior to mistake-seeding. Since the textbook is widely used by educators for introductory computer science theory courses, we are interested in evaluating the efficacy of the presented material for student outcomes, and as such, do not alter the proof to make the prose or logic more or less comprehensible. Since figures are frequently used as an educational instrument (e.g., [52]), we included, with each stimulus, a figure related to that formalism comprehension task taken from the textbook or instructor slides for the theory courses. Figure 2 shows a sample stimulus for the Towers of Hanoi problem.

We make all of our stimuli publicly available in our replication package.

C. Experimental Protocol

We recruited participants via in-class invitations and online class discussion forums. Participants were asked to read and sign the general consent form prior to their scheduled 60-minute experimental session. Each session had three compo-

nents: pre-questionnaire survey, eye-tracking session, and post-questionnaire survey and debriefing.

Pre-Questionnaire Survey. After participants re-affirmed their consent, we administered a survey to collect basic demographic data (e.g., gender, native language, class standing, etc.). To measure the incoming preparation for participants, we collected data on the theory-related courses they have completed or are currently enrolled in, and asked participants to complete a screening question from a widely-used undergraduate discrete mathematics textbook [22].

Eye-Tracking Session. Participants were seated in front of a computer screen with a Tobii Pro X3-120 eye tracker in a quiet room with controlled ambient light and screen brightness levels. Participants were encouraged not to look away from the computer screen to help reduce noise in the gaze and facial behavior data. We first calibrated the eye-tracker for each participant. We then showed the participants training slides explaining the study design and purpose. Participants were informed they would be reading several algorithmic proofs from an undergraduate textbook and determining whether or not each proof is correct. Each participant was presented with 7 stimuli. All stimuli were presented within the interface provided by Tobii Pro Lab [53], and participants selected their answers via key presses.

Post-Questionnaire Survey. After the eye-tracking session, we instructed participants to complete a post-questionnaire survey and asked them to self-report (on a 1–5 Likert scale) their prior perceived experience with computer science formalisms, difficulty of tasks they were asked to perform, and helpfulness of different aspects of the formalism presentation. Note that while it is common to ask participants to self-report their experience prior to the start of the experiment, we include such questions in the post-questionnaire to mitigate potential decrease in performance due to stereotype threat [54]. This is especially relevant for pedagogy since stereotype threat is reported to disproportionately affect underrepresented groups and students with non-traditional backgrounds [55] that may already struggle with outcomes in theory courses. In addition to the Likert scale data, we also collected qualitative responses from participants on attributes that make a formalism comprehension task easier to complete.

D. Data Collection

We conducted all experiments on a 64-bit Windows 10 machine connected to a 27 inch monitor with a 1920x1080 resolution. To collect eye gaze data, we used the Tobii Pro X3-120, a non-invasive eye tracker that can detect fixations at the granularity of a single line of 10pt text. Our eye tracker was set to sample readings at a frequency of 120hz (i.e., 120 times per second). We processed this raw data using Tobii Pro Lab to generate analyzable gaze data.

To collect facial behavior data, we used the Logitech C920 webcam. Our webcam was set to record 1080p video at 30 frames per second. The high-definition recorded video was then processed offline using OpenFace [56], an open source toolkit capable of automatically detecting the presence of facial AUs and reporting degrees of confidence.

IV. Data Analysis Approach

In this section, we present the mathematical analyses applied to our eye-tracking and facial behavior data. We applied a *false discovery rate* (FDR) threshold at $q < 0.05$ to correct for multiple comparisons (i.e., to avoid false positives as a result of repeated analyses). All reported measures of statistical significance in Section V correspond to p-values corrected for multiple comparisons.

A. Eye-Tracking Analysis Approach

To preprocess for data quality, we filter outlier data points by removing the responses that were keyed in too quickly (outside $1.5\times$SD of the mean response time) and therefore, could not reasonably correspond to the participants reading a formalism presentation entirely before selecting an answer. We also filter out data points that correspond to noisy gaze data [17, Sec. 7.1]. This filtering resulted in 191 out of the original 236 data points being usable for experimental analyses.

Following the Goldberg and Helfman guidelines [32] for defining AOIs in terms of size and granularity, we manually divide each presented stimulus into six AOIs: *Algorithm*, *Theorem*, *Proof*, *Figure*, *Correct Answer*, and *Distractors*. The *Algorithm* AOI represents the pseudocode algorithm of interest, including the inputs and outputs of the algorithm and any explanatory comments. The *Theorem* and *Proof* AOIs represent the prose text for the theorem and the proof respectively. The *Figure* AOI corresponds to a graphical illustration of the formalism comprehension task and includes relevant captions and labels. Finally, the *Correct Answer* and *Distractors* AOIs represent the multiple choice responses.

We analyzed raw eye-movement data to detect velocity-based fixations (I-VT) [57], a commonly-used fixation extraction method in the research community [58]. We use the following standard metrics to analyze and compare the strategies employed by participants for the formalism comprehension tasks. A *strategy* models gaze data and visual attention trends over time for the duration of a task. The *fixation time* corresponds to the total duration of all fixations on an AOI, while the *fixation count* indicates the total number of fixations on an AOI. Longer fixation times indicate either higher levels of interest or increased difficulty, and as such, increased strain on the working memory, in extracting information from the AOI [30, 59]. The *regression rate* depends on saccadic eye movements and indicates the percentage of backward saccades [60], and higher regression rates indicate increased difficulty in completing a task [35]. The *attention switching* metric depends on fixation counts and measures the total number of switches between AOIs, and can approximate the dynamics of visual attention during a task [30].

Previous work has argued for the use of baseline pupil diameters [39], and we used the training slides administered after eye-tracking calibration to measure the baseline pupil

TABLE I: The facial Action Units (AUs) in our study

AU	Description	AU	Description
1	Inner brow raiser	14	Dimpler
2	Outer brow raiser	15	Lip corner depressor
4	Brow lowerer	17	Chin raiser
5	Upper lid raiser	20	Lip stretcher
6	Cheek raiser	23	Lip tightener
7	Lid tightener	25	Lips part
9	Nose wrinkler	26	Jaw drop
10	Upper lip raiser	28	Lip suck
12	Lip Corner Puller	45	Blink

diameter (and as such, approximate the cognitive load) for participants prior to working on the formalism comprehension tasks.

B. Facial Behavior Analysis Approach

We use the open-source toolkit OpenFace [56] to analyze recorded webcam video from participants and automatically detect the presence of facial action units (AUs).

During the preprocessing stage, since we intend to use facial behavior analysis as exploratory research and wish to mitigate threats to conclusion validity (given the lack of widely-accepted metrics for analyzing facial behavior data [48, 49]), we only use measurements reported by the OpenFace classifier with the highest possible confidence value (98%). Additionally, we also filter out facial behavior data for any timestamps not corresponding to a fixation on an AOI in the gaze data, since we wish to analyze facial behavior for different study outcomes. After the preprocessing step, we are left with around 24 thousand high-quality data points (out of about 1.7 million original) for facial behavior analysis.

We analyze facial data "as is" in terms of AUs without attempting to link it to emotions. Using facial behavior data to classify human emotions has been widely studied [42, 43, 44, 45] but remains controversial, given that emotions are highly subjective and depend on the context [48, 49]. Even a categorization of the valence of emotions (i.e., positive and negative) has been met with skepticism by the research community recently. By contrast, facial AUs remain an effective measure of the facial behavior, since they report the objective physiological behavior of an individual's facial muscles, not obfuscated by machine-learned emotion classifications trained on certain population demographics. As such, we consider only the high-confidence values of 18 different facial AUs reported by OpenFace (see Table I for the official anatomic description of each AU [47]). To encourage conversations about facial behavior analysis in the context of pedagogy, we discuss exploratory results from our facial behavior analysis in Section VI-B. We also make our de-identified facial behavior data publicly available for researchers to replicate and build on our work.

V. Experimental Results

We consider the following research questions:

RQ1. What is the effect of incoming preparation on student outcomes for formalism comprehension tasks?

RQ2. How do student self-reports of formalism comprehension tasks align with empirical results?

RQ3. What factors most distinguish higher-performing individuals from lower-performing ones?

Table II outlines the independent and dependent variables for each RQ, including the metrics used for each variable. Explanations of key terms and eye-tracking metrics in a software engineering context follow in the relevant RQ subsections.

A. RQ1. Role of Incoming Preparation

We examine the relationship between incoming preparation of participants and outcomes for the formalism comprehension tasks. We consider two facets of preparation: coursework count and performance. First, for *coursework count*, we enumerate the number of computer science theory courses covering formalisms (i.e., courses that include proofs and derivations in their syllabi) that participants have either completed with passing grades or are currently taking. Second, for *screening proof performance*, we asked participants to identify a mistake in a proof distinct from the stimuli used in the study, and note whether the participant correctly identified the mistake. Both facets we consider have been used previously in the context of pedagogy to approximate incoming preparation [23, 25]. While we note that factoring in grades for the theory courses would result in a more accurate approximation of incoming preparation, instructors for upper-level courses typically only require a student to pass the prerequisite courses, and do not know how well students did in the core courses. As such, we do not consider grades as a proxy for incoming preparation. We classify a participant who has taken above the median number of theory courses (i.e., coursework count > 4 for our dataset) and passes the screening question as *more-prepared*. Applying our approximations for incoming preparation resulted in 16 out of 34 participants being classified as more-prepared, with the remaining 18 deemed less-prepared.

The mean *response accuracy* (i.e., percentage of correct answers) and *response time* (i.e, time taken to choose an answer) for more-prepared and less-prepared participants is shown in Table III. Surprisingly, we found no evidence of a statistically-significant difference in the outcomes between more-prepared and less-prepared students, both in terms of response accuracy (two-tailed Mann-Whitney U-test with the Benjamini-Hochberg [61] procedure to correct for false discoveries, $p = 0.96$) and response time ($p = 0.93$). Notably, while absence of evidence is not evidence of absence, the effect sizes for both results were extremely small (Cohen's $d = 0.007$ for response accuracy and $d = 0.08$ for response time), giving statistical confidence in the null result (i.e., even if an effect were present, it would be of very low magnitude and thus unlikely to influence outcomes).

We further found no correlation between the number of theory courses taken and response accuracy (Pearson's $r = 0.036, p = 0.84$), nor a correlation between participants' self-perceived experience with formalisms (on a 1–5 Likert scale) and response accuracy (Kendall's $\tau = 0.21, p = 0.18$). Our results indicate that, contrary to conventional wisdom [23, 24,

TABLE II: Independent and dependent variables for each RQ, along with associated metrics. Descriptions of key terms follow in the relevant RQ subsections.

	Independent Variables	Metrics: Independent Variables	Dependent Variables	Metrics: Dependent Variables
RQ1	Incoming preparation	Coursework count, screening proof performance	Task performance	Response times and accuracy
			Visual attention	Fixation times on AOIs
			Task difficulty	Regression rates for proof types
RQ2	Self-perceived experience	Likert scale	Task performance	Response accuracy
	Self-perceived task difficulty			
	Self-perceived helpfulness of figures			
	Self-perceived proof readability			
	Visual attention to pseudocode	Fixation time		
	Visual attention to figures and proofs			
RQ3	Proof type	Categorical (inductive, contradictive, and direct proofs)	Task performance	Response accuracy
	Algorithm type	Categorical (recursive, iterative, and non-repeating algorithms)		
	Visual behavior	Attention switching count		
	Cognitive load	Pupil diameter		

TABLE III: Mean response accuracy, response time, and fixation times (FT) on different AOIs for more-prepared and less-prepared participants. Response and fixation times are given in seconds, while response accuracy is shown as a percentage.

	Mean (SD)		
	More-prepared	Less-prepared	p
Response Time	248.7(±109.6)	240.0(±105.3)	0.93
Response Accuracy	34.8(±17.3)	32.5(±16.1)	0.96
Algorithm FT	19.5(±16.2)	17.1(±17.4)	0.21
Correct Answer FT	**1.3(±2.0)**	**0.8(±1.2)**	**0.038**
Distractor Choices FT	**14.6(±12.3)**	**11.1(±11.5)**	**0.03**
Figure FT	11.0(±45.8)	10.9(±38.6)	0.88
Proof FT	**66.8(±12.3)**	**46.1(±11.5)**	**0.005**
Theorem FT	12.9(±2.0)	11.2(±1.2)	0.12

25, 26] and instructor expectations, students with greater incoming preparation perform no better at these formalism tasks, on average, than students with lower incoming preparation. Indeed, the two participants with the highest number of courses taken had the lowest and second-lowest response accuracies.

These results have potentially major implications, both for pedagogy and the training of new hires for formal software engineering. On the pedagogy front, our results raise questions about course design and undergraduate curricula: upper-level undergraduate courses are often designed with the expectation that students will have completed, and will be familiar with, material covered in core courses. If students with more exposure to the formal material do not show evidence of retention over time, educators may need to reconsider upper-level course design with more of an emphasis on reviewing relevant material covered in lower-level courses. For high-assurance software engineering, some managers may be tempted to make hiring and training decisions based on the number of theory courses taken (e.g., from a transcript or resume list). Our results add confidence that regardless of the number of relevant courses taken, new hires for high-assurance software engineering should be put through the same level of training to ensure that they are prepared for the challenges of the job, and that managers should not default to "courses completed" as a proxy of preparation for the job.

Even though participants have similar final outcomes, they employ different strategies. An analysis of visual behaviors between students with different incoming preparations reveals that more-prepared students *fixate longer* on (i.e., spend more time looking at) AOIs corresponding to the proof (two-tailed Mann-Whitney U-test with the Benjamini-Hochberg procedure, $p = 0.005$), correct answer ($p = 0.038$), and distractor answer choices ($p = 0.03$). The mean fixation times for all six AOIs for more- and less-prepared participants are included in Table III. Our results suggest that while incoming preparation teaches students to read a proof and the answer choices thoroughly before selecting an answer, this increased attention to the AOIs does not actually help students achieve better outcomes.

Recall that *regression rate* is the ratio of backward or regressive saccades to the total number of saccades. Quite surprisingly, we observed that students with greater incoming preparation show a higher regression rate (i.e., informally, spend more time re-reading text and figures) — and as such, increased difficulty [35] — while reading direct proofs ($p = 0.035$). Given that we found no evidence of a statistically-significant difference in the performance of students with different incoming preparation for direct proofs, our results suggest that, for our sample, students may be trained in theory courses to default to induction or contradiction as proof strategies, and may need to put in more mental effort when analyzing a direct proof — a style that remains highly relevant in formal methods for software engineering (e.g., [8]).

We found no evidence that students with higher incoming preparation, as traditionally assessed, perform better at formalism comprehension tasks ($p = 0.96$). This suggests the need for pedagogical intervention in core theory courses to ensure student outcomes are aligned with course objectives of having students master the material and better preparing them for formal methods in software engineering.

B. RQ2. Self-Reporting and Formalism Comprehension Tasks

To collect richer free-response data from our study, we instructed all 34 participants to provide answers to a post-questionnaire reflecting on their experiences with the study and outlining what they thought to be the most important parts of a formalism presentation. In addition to having participants self-report (on a 1–5 Likert scale) the difficulty of tasks they were asked to perform, the helpfulness different aspects of the formalism presentation, and so on, we asked three free-response questions:

1) Having completed the study session, would you do anything different the next time around?
2) What is the most important thing that makes a proof easier to understand?
3) What is the most important thing that makes it easier to spot a mistake in the proof?

To better understand the relationship between participant Likert scale responses and *response accuracy*, we use the Kendall's τ test to conduct a quantitative analysis of the data. When analyzing participants' *self-reported experience* with formalisms, we found no evidence of a correlation between experience and response accuracy ($\tau = 0.21, p = 0.18$). We also observed no correlation between *self-reported task difficulty* and study outcomes ($\tau = 0.14, p = 0.35$), nor a correlation between the *self-reported helpfulness of figures* for formalism comprehension tasks and study outcomes ($\tau = -0.22, p = 0.13$). Our results do not provide evidence that students are accurate at self-reporting their experience with formalism comprehension tasks.

To further investigate whether student self-perception is an accurate predictor of factors associated with high outcomes, we also performed a qualitative analysis on the participants' self-reported free-response data. 26 out of 34 participants indicated they would employ a different problem-solving strategy if asked to do the study again. Tied for the most common change in strategy were paying more attention to the algorithmic pseudocode and reviewing the materials from core theory courses prior to the study. Our experimental results, on the other hand, do not indicate a relationship between *fixation time* on algorithmic pseudocode (i.e., time spent reading pseudocode) and higher response accuracy ($p = 0.91$, see Section V-C). The desire to review materials from core theory courses is aligned with our experimental results: students with greater incoming preparation do not perform better, suggesting a lack of retention of course materials over time, and hence, preparation for high-assurance software.

When asked to describe the features that make a proof easier to comprehend, about a third of the participants mentioned concise, easy-to-read English prose in the proof. The second most popular answer (6/34 participants) corresponded to the use of figures and visuals while reading the proof. Interestingly, our empirical results do not show evidence of a significant correlation between *self-perceived proof readability* and outcomes for formalism comprehension tasks (Kendall's $\tau = -0.14, p = 0.32$), or a statistically significant relationship between increased *fixation* on (or attention to) figures and response accuracy ($p = 0.81$).

In response to the traits that make a mistake in a proof easier to spot, the most popular answer (7/34 participants) focused on step-by-step logical reasoning. The second most common answer themes (6/34 participants each) were logical inconsistencies in the proof text and an understanding of the proof strategy. By contrast, only one participant answered "thinking through a different worked example", a strategy that is commonly taught in undergraduate theory courses. Our results suggest that educators should put more of an emphasis on providing students with effective tools for evaluating logical deductions for correctness. The student-perceived traits (i.e., breaking down logical reasoning steps and evaluating the reasoning for logical inconsistencies) remain effective strategies for formalism comprehension tasks. However, the lower outcomes for these tasks suggests that students are less able to apply those strategies in a mistake-finding context.

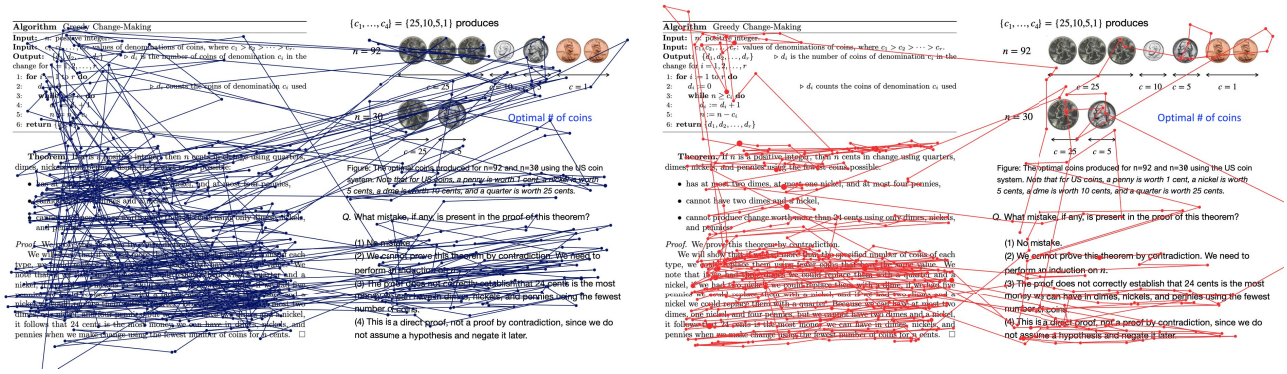
> We find no evidence that students experience reports are accurate predictors of outcomes for comprehending formalisms ($\tau = 0.21, p = 0.18$). We also find no evidence that the factors identified by students are associated with high outcomes for such tasks.

C. RQ3. Factors Associated with Higher Outcomes

Given the apparent lack of effect of incoming preparation on the outcomes for our study, we are interested in investigating the factors that cause students to perform better at formalism comprehension tasks. To do so, we perform a sub-population analysis of students with higher and lower outcomes. We require a participant to have achieved above the median response accuracy (i.e., $\geq 3/7$ answers correct for our dataset) to be classified as higher performing. Using this metric, we classify 15 out of 34 participants as *higher performing*, with the remaining 19 considered *lower performing* participants. Only 7 out of the 16 more-prepared participants (Section V-A) were classified as higher performing.

We examined the outcomes for higher and lower performing students for different proof categories (inductive, contradictory, and direct) and algorithm categories (recursive, iterative, and non-repeating[1]). We found that, independent of algorithm category, higher performing students are more likely to spot mistakes in proofs by induction than the lower performing ones (χ^2 test with the Benjamini-Hochberg procedure, $p = 0.01$). Endres *et al.* [62] have investigated student performance for iterative and recursive problem formulations, and observed poorer student performance for recursive algorithms involving non-branching computation but better student performance for recursive algorithms involving array manipulation. We find that independent of problem or proof type, higher performing students are more likely to get proofs for recursive algorithms correct compared to lower performing students ($p = 0.006$).

[1]The only algorithm in our stimuli that does not involve loops or recursion corresponds to a proof gadget used for the Halting Problem. For completeness, we consider "non-repeating" a separate category of algorithms.



(a) Higher-outcome participant gaze plot

(b) Lower-outcome participant gaze plot

Fig. 3: Visual gaze plots for a stimulus shown to two participants. The higher-outcome participant (left) displays significantly more attention switching behaviors, as indicated by the number of lines crossing between different AOI quadrants.

Our results have implications for both teaching formalisms for improved outcomes and preparing students for formal methods in software engineering. Previous work by Polycarpou [63] suggests that students who understand recursive or inductive definitions can more successfully perform proofs by induction, while students who do not are either not able to perform proofs by induction, or do so mechanically. The fact that students with higher outcomes in our study are not necessarily those with greater incoming preparation suggests that undergraduate theory courses taken by our participants may not be putting emphasis on teaching and making students comfortable with recursive or inductive definitions. In formal verification for high-assurance software, there has been an increasing interest in automated theorem proving (e.g., the Z3 theorem prover [64]), an activity that often involves the identification of an *inductive invariant* to prove that a certain property holds at all time [65]. Finding an inductive invariant has a direct parallel to correctly identifying an inductive hypothesis for proofs by induction, suggesting that students who are better trained to correctly establish inductive proofs may be better equipped for automated theorem proving tasks.

Recall that *attention switching* measures the total number of switches between AOIs, and can approximate the dynamics of visual attention during a task. We found that higher performing students demonstrate more *attention switching* behaviors, or frequently go back and forth between AOIs on the presented materials (two-tailed Mann-Whitney U-test with the Benjamini-Hochberg procedure, $p = 0.002$). In particular, we observed a statistically-significant difference in attention switching for proofs by contradiction ($p = 0.009$) and iterative algorithms ($p = 0.007$). We also observed trends for increased attention switching for higher-performing students for proofs by induction and recursive algorithms, but these trends did not survive correcting for multiple comparisons. Figure 3 shows two illustrative visual gaze plots for a higher outcome and another lower outcome participant for a stimulus involving proof by contradiction. The higher-outcome participant displays significantly more attention switching behavior (as indicated by the increased number of lines between the AOI quadrants, see Figure 3a) compared to the lower-outcome participant (Figure 3b). In the context of pedagogy, these results strongly argue for the development of teaching materials, such as online tools, lecture slides, and exams, that facilitate perusal with ease (e.g., without requiring multiple page flips).

To estimate the *cognitive load* (i.e., increased strain on the working memory) due to formalism comprehension tasks (e.g., due to the need for remembering the theorem or variable names while evaluating the proof for correctness), we record the pupil diameters reported by the eye tracker for each stimulus and obtain the pupil diameter delta against a measured baseline (see Section IV-A). We found that students with poorer performance also show increased cognitive load when going over the figures in general ($p = 0.012$). In particular, we saw trends for increased cognitive load for lower outcome students looking at figures for inductive proofs and recursive algorithms, though the latter did not survive correcting for multiple comparisons ($p = 0.032$ and $p = 0.06$ respectively). These results indicate that students who got inductive proofs incorrect more frequently (in a statistically-significant manner) also had increased difficulty when going over figures explaining the algorithm or proof strategy when compared to their higher-performing peers, suggesting that educators can make targeted efforts to have lower-outcome students practice tracing through graphical illustrations of the presented material.

> Higher-performing students are more likely to get proofs by induction ($p = 0.01$) and recursive algorithms correct ($p = 0.006$) compared to lower-performing students. Higher-outcome students also demonstrate significantly more attention switching behaviors ($p = 0.002$), suggesting that students who frequently go back and forth between presented materials are more likely to achieve better results.

VI. Discussion and Implications

In this section, we present a discussion of our results and discuss several implications for pedagogy (and indirectly,

preparing future software engineers for formal methods).

A. Eye-tracking Discussion and Implications for Pedagogy

The results from our study suggest that traditional metrics for incoming preparation, like course counting and pretests, are ineffective predictors of student performance with formalism comprehension tasks, and that students across the board are not well-trained to employ different tools for evaluating presented logical deductions for correctness. Indeed, the two most-prepared participants in our study had the lowest and second-lowest response accuracies, and the highest-outcome participants were more junior.

Our results provide confidence that students with more exposure to the formal material may not show evidence of retention over time. As such, it may benefit students if educators focus on upper-level course design strategies that encourage reviewing relevant material from lower-level undergraduate courses. The trade-off between using a few lectures to ensure students who took core courses several semesters ago still remember key concepts and exposing students to novel topics is worth considering.

Given that higher-outcome participants exhibited significantly more attention switching, we consider whether current educational materials admit this sort of problem-solving strategy. For instance, our results argue against exams that require page flips to get relevant information, since turning or scrolling between pages is not conducive to going back and forth between presented materials with ease (e.g., [66, Sec. 6.2.1]). In an era of an increasing number of online exams and teaching tools, similar concerns arise: while administering online lectures, quizzes, and exams, educators should remain wary of requiring significant page scrolling or user interface navigation to gather information before answering a question, and instead place relevant bits of information in a spatially-proximate manner for a formalism comprehension task.

Additionally, since we observe differences in performance depending on the type of algorithm or proof, we advocate for increased emphasis on certain types of proofs. Notably, our study shows that differences in outcomes can be attributed, in a statistically-significant manner, to proofs by induction. Previous work has shown that students' performance with proofs by induction improves after class instruction, but not to the extent intended during course design [63], yet strategies involving proofs by induction remain highly relevant in formal verification of software (e.g., the use of inductive invariants in automated theorem proving [65]). We encourage educators to have students practice proofs by induction more, and emphasize familiarity and comfort with recursive or inductive definitions and data structures.

Note that while motivated by our experimental results, our recommendations for educators (and, more indirectly, hiring managers) are speculative and only intended for discussion purposes: a controlled study would be required to assess any interventions. We leave such studies for future work.

B. Facial Behavior Analysis Exploratory Results

We wish to analyze any differences in facial behavior between participant interactions with different outcomes. We note that, to the best of our knowledge, no widely-accepted metrics for analyzing facial behavior data in the context of pedagogy exist (e.g., previous use of facial behavior analysis in a teaching setting [44, 45] involves the controversial use of emotion detection [48, 49, 67]). We perform a subpopulation analysis of facial behavior data points corresponding to participant correct and incorrect answers. 56% of the 24 thousand high-quality facial behavior data points correspond to participant interactions getting the response correct, with the remaining associated with incorrect answers. For each AU (see Table I), we analyze the number of times the OpenFace classifier detected its presence or absence. We compare this AU activation metric for correct vs. incorrect interactions (χ^2 contingency test, corrected for false discoveries using the Benjamini-Hochberg procedure). Our exploratory results indicate a statistically-significant difference between participant interactions for different response accuracies for 16 of the 18 AUs; only AU7 (lid tightener) and AU9 (nose wrinkler) were not relevant. Differences in all but two AUs suggest an easy-to-observe, robust effect.

While our exploratory results suggest the presence of statistically-significant differences in many different facial behaviors between participant interaction outcomes, there unfortunately remains a lack of metrics and analysis (other than generic emotion classifications — see Section IV-B) methods to help us interpret these results. A qualitative theory to explain this difference in facial behavior is left as future work. We present this discussion on the preliminary facial behavior analysis results in the context of formalism comprehension tasks to draw attention of the research community to a more objective method of facial behavior analysis (that is also observable and robust in this domain).

VII. Threats to Validity

One threat to validity for our study is that our results may not generalize to a wider population. To mitigate this threat, we recruited participants from a large public university with a wide array of different backgrounds (including native language, incoming preparation, class standings, etc.). We also note that the primary goal of our study is to understand how to better teach formalisms at the undergraduate level (and indirectly, to shed light on hiring considerations for certain software engineering sectors), and thus, recruiting seasoned industry professionals is less relevant.

Another threat to the generalizability of our study is that our stimuli may not be indicative (e.g., all of the proofs are in English). To mitigate this threat, we select our stimuli from an undergraduate textbook widely used by educators, and supplement any figures from undergraduate course lecture slides serving thousands of students each year.

Finally, to mitigate threats to conclusion validity, we use state-of-the-art software to calibrate the eye-tracker [53], widely-used eye-tracking metrics and analyses [30], and

present any facial behavior analysis conclusions as discussions meant to stimulate conversations in the research community.

VIII. Related Work

In this section, we discuss previous work related to eye-tracking, student cognition, and facial behavior analysis in a pedagogical context.

A. Eye-tracking and Cognition

Previous eye-tracking work investigating problem-solving strategies employed by students has shown that differences in search strategies can lead to significant differences in task outcomes [68, 69, 70].

For instance, Netzel *et al.* found that high-accuracy students were better able to use information in science-related diagrams [68]. We similarly observed that high-performing participants display reduced cognitive load when going over figures related to formalism comprehension tasks. Hegarty *et al.* [69] investigated arithmetic problem solving involving relational terms inconsistent with the required arithmetic operator (e.g., the use of *less than* for tasks involving addition), and found that low-accuracy students made more reversal errors for inconsistent problems, and that high-accuracy ones required more re-readings for previous text fixations. Our study does not reveal a difference in response accuracy between high- and low-outcome students for proofs by contradiction (that involve logical inconsistencies in proof text). We do find, however, that higher-outcome students display significantly more attention switching as they assimilate presented information, a strategy that is analogous re-reading text.

Figures are frequently used as educational instruments [52, 71, 72], yet their importance as a medium of instruction for a particular field is not always well-understood. For instance, Susac *et al.* [70] found that diagram were rarely helpful for physics. By contrast, Yoon *et al.* [73] studied the importance of figures for causal reasoning problems, and found that even for questions missing a figure, 48% of the students still frequently fixate on the area where the figure would have been, indicating a relative higher importance for figures. For both studies, however, the inclusion of diagrams did not affect the participants' time taken to respond or response accuracy. In a mistake-finding context for formalism comprehension, we similarly do not find a relationship between fixation on, or perceived importance of, figures on task outcomes.

B. Facial Behavior Analysis

Previous work exploring facial behavior analysis in a pedagogical setting has analyzed videos of faces as they interact with a teaching tool to assign general notions of affective states (e.g., boredom, delight, surprise, etc.) during learning tasks [44, 45, 74]. Such studies argue that highly-animated affective states or emotions, such as delight or confusion, are easily detectable and have numerous applications for pedagogy, including real-time feedback for learners [44].

Crucially, while these studies use AUs as measures of facial behavior, their assignment of affective states or emotions is based on generic classifiers [75] and fails to account for the context around the task, a feature considered imperative for emotion recognition [48, 49]. Additionally, Alfenbein and Ambady [76] show that the accuracy of facial emotion recognition can depend on race and culture. As such, we argue that emotion classifiers trained on certain population demographics can inadvertently affect vulnerable student populations if used without caution in a pedagogical setting.

We present our exploratory results from facial behavior analysis independent of emotion classification to promote discussion in the research community for a less controversial use of facial behavior data that could still yield benefits similar to those of previous tools like AutoTutor [75] without potentially negatively affecting minority or non-traditional students.

IX. Conclusion

Formal methods have been increasingly applied to software engineering, but often require mathematical training and advanced logical reasoning abilities that software practitioners often do not possess. Given the challenges associated with integrating formal methods into software, educators may increasingly focus on formalisms in undergraduate theory courses that already suffer from unsatisfactory student outcomes.

We propose to use eye-tracking to better understand the problem solving strategies employed by students with different levels of incoming preparation and task outcomes, and more indirectly, gather insights into how educators can prepare future software engineers for the rigorous logical reasoning that is a core part of high-assurance software engineering. We also provide an exploratory discussion on the use of facial behavior analysis in a pedagogical context.

In a controlled human study involving 34 participants, we find that incoming preparation is not an accurate predictor of task outcomes, that student experience reports and self-perceptions are not effective at predicting task outcomes, and that the increased attention to proof text by more-prepared students does not yield higher task outcomes. We instead find that students who exhibit more attention switching behaviors are more likely to succeed, and that differences in formalism comprehension outcomes can be attributed to performance for proofs by induction and recursive algorithms. Our results advocate for pedagogical interventions in theory courses to better teach formalisms to students and prepare future developers for formal reasoning for software. We make our datasets publicly available for researchers to replicate or build on our study.

Acknowledgements

We gratefully acknowledge the partial support of the NSF (2211749, 1908633) and the AFRL (FA8750-19-1-0501) as well as the University of Michigan Center for Academic Innovation. Additionally, we thank Madeline Endres, Wenxin He, Priscila Santiestaban, and Amaris Sim for their logistical help and their help piloting the stimuli.

REFERENCES

[1] E. M. Clarke and J. M. Wing, "Formal methods: State of the art and future directions," *ACM Computing Surveys*, vol. 28, no. 4, pp. 626–643, 1996.

[2] V. George and R. Vaughn, "Application of lightweight formal methods in requirement engineering," *STSC CrossTalk–The Journal of Defense Software Engineering*, 2003.

[3] B. Meyer, "Applying'design by contract'," *Computer*, vol. 25, no. 10, pp. 40–51, 1992.

[4] C. B. Jones, "Systematic software development using vdm," *Prentice Hall International Series in Computer Science*, 1990.

[5] R. W. Floyd, "Assigning meanings to programs," in *Program Verification*. Springer, 1993, pp. 65–81.

[6] C. A. R. Hoare, "An axiomatic basis for computer programming," *Communications of the ACM*, vol. 12, no. 10, pp. 576–580, 1969.

[7] R. M. Hierons, K. Bogdanov, J. P. Bowen, R. Cleaveland, J. Derrick, J. Dick, M. Gheorghe, M. Harman, K. Kapoor, P. Krause *et al.*, "Using formal specifications to support testing," *ACM Computing Surveys*, vol. 41, no. 2, pp. 1–76, 2009.

[8] T. Hoare, "Assert early and assert often: Practical hints on effective asserting," *Presentation at Microsoft Techfest*, 2002.

[9] C. Heitmeyer, "On the need for practical formal methods," in *International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems*. Springer, 1998, pp. 18–26.

[10] M. Gleirscher, S. Foster, and J. Woodcock, "New opportunities for integrated formal methods," *ACM Computing Surveys*, vol. 52, no. 6, pp. 1–36, 2019.

[11] S. Liu, K. Takahashi, T. Hayashi, and T. Nakayama, "Teaching formal methods in the context of software engineering," *ACM SIGCSE Bulletin*, vol. 41, no. 2, pp. 17–23, 2009.

[12] J. Kloosterman and D. Fontenot, *Comprehensive Studies Program (CSP) Support Planning Discussions AY 2019-2020*, 2020, university of Michigan meeting held on 08/04/2020.

[13] V. Bertacco and A. Kamil, *Computing CARES Survey AY 2019-2020*. University of Michigan, 2020.

[14] J.-R. Abrial, "Formal methods in industry: achievements, problems, future," in *Proceedings of the 28th international conference on Software engineering*, 2006, pp. 761–768.

[15] S. Fakhoury, Y. Ma, V. Arnaoudova, and O. Adesope, "The effect of poor source code lexicon and readability on developers' cognitive load," in *International Conference on Program Comprehension (ICPC)*. IEEE, 2018, pp. 286–28 610.

[16] N. Peitek, J. Siegmund, C. Parnin, S. Apel, J. C. Hofmeister, and A. Brechmann, "Simultaneous measurement of program comprehension with fmri and eye tracking: A case study," in *International Symposium on Empirical Software Engineering and Measurement*, 2018, pp. 1–10.

[17] Z. Sharafi, I. Bertram, M. Flanagan, and W. Weimer, "Eyes on code: A study on developers code navigation strategies," *IEEE Transactions on Software Engineering*, pp. 1–1, 2020.

[18] R. Krueger, Y. Huang, X. Liu, T. Santander, W. Weimer, and K. Leach, "Neurological divide: An fMRI study of prose and code writing," in *International Conference on Software Engineering*, ser. ICSE '20, 2020, p. 678–690.

[19] Y. Huang, X. Liu, R. Krueger, T. Santander, X. Hu, K. Leach, and W. Weimer, "Distilling neural representations of data structure manipulation using fMRI and fNIRS," in *International Conference on Software Engineering*, 2019, pp. 396–407.

[20] Y. Huang, K. Leach, Z. Sharafi, N. McKay, T. Santander, and W. Weimer, "Biases and differences in code review using medical imaging and eye-tracking: Genders, humans, and machines," in *Foundations of Software Engineering*, 2020, p. 456–468.

[21] J. Mock, S. Huber, E. Klein, and K. Moeller, "Insights into numerical cognition: Considering eye-fixations in number processing and arithmetic," *Psychological Research*, vol. 80, no. 3, pp. 334–359, 2016.

[22] K. H. Rosen, *Discrete mathematics and its applications*, 7th ed. McGraw-Hill, 2012.

[23] S. Salehi, S. Cotner, and C. J. Ballen, "Variation in incoming academic preparation: Consequences for minority and first-generation students," in *Frontiers in Education*, vol. 5. Frontiers Media SA, 2020, p. 552364.

[24] C. A. Stanich, M. A. Pelch, E. J. Theobald, and S. Freeman, "A new approach to supplementary instruction narrows achievement and affect gaps for underrepresented minorities, first-generation students, and women," *Chemistry Education Research and Practice*, vol. 19, no. 3, pp. 846–866, 2018.

[25] P. A. Tolley, C. M. Blat, C. R. McDaniel, D. B. Blackmon, and D. C. Royster, "Enhancing the mathematics skills of students enrolled in introductory engineering courses: Eliminating the gap in incoming academic preparation," *Journal of STEM Education: Innovations and Research*, vol. 13, no. 3, 2012.

[26] J. R. Reisel, M. Jablonski, H. Hosseini, and E. Munson, "Assessment of factors impacting success for incoming college engineering students in a summer bridge program," *Intl. J. of Mathematical Education in Sci. and Tech.*, vol. 43, no. 4, pp. 421–433, 2012.

[27] T. Strandvall, "Eye tracking in human-computer interaction and usability research," in *IFIP Conference on Human-Computer Interaction*. Springer, 2009, pp. 936–937.

[28] U. Obaidellah, M. Al Haek, and P. C.-H. Cheng, "A survey on the usage of eye-tracking in computer programming," *ACM Computing Surveys*, vol. 51, no. 1, pp. 1–58, 2018.

[29] Z. Sharafi, Z. Soh, and Y.-G. Guéhéneuc, "A systematic literature review on the usage of eye-tracking in software engineering," *Information and Software Technology*, vol. 67, pp. 79–107, 2015.

[30] Z. Sharafi, T. Shaffer, B. Sharif, and Y.-G. Guéhéneuc, "Eye-tracking metrics in software engineering," in *Asia-Pacific Software Engineering Conference*, 2015, pp. 96–103.

[31] M. Wedel and R. Pieters, "A review of eye-tracking research in marketing," *Review of marketing research*, pp. 123–147, 2017.

[32] J. H. Goldberg and J. I. Helfman, "Comparing information graphics: a critical look at eye tracking," in *BEyond time and errors: novel evaLuation methods for Information Visualization*, 2010, pp. 71–78.

[33] Z. Sharafi, B. Sharif, Y.-G. Guéhéneuc, A. Begel, R. Bednarik, and M. Crosby, "A practical guide on conducting eye tracking studies in software engineering," *Empirical Software Engineering*, vol. 25, no. 5, pp. 3128–3174, 2020.

[34] M. A. Just and P. A. Carpenter, "A theory of reading: from eye fixations to comprehension." *Psychological review*, vol. 87, no. 4, p. 329, 1980.

[35] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *Journal of industrial ergonomics*, vol. 24, no. 6, pp. 631–645, 1999.

[36] K. Rayner, "Eye movements in reading and information processing: 20 years of research." *Psychological bulletin*, vol. 124, no. 3, p. 372, 1998.

[37] M. A. Eskenazi and J. R. Folk, "Regressions during reading: The cost depends on the cause," *Psychonomic bulletin & review*, vol. 24, no. 4, pp. 1211–1216, 2017.

[38] S. Chen and J. Epps, "Using task-induced pupil diameter and blink rate to infer cognitive load," *Human–Computer Interaction*, vol. 29, no. 4, pp. 390–413, 2014.

[39] J.-L. Kruger, E. Hefer, and G. Matthew, "Measuring the impact of subtitles on cognitive load: Eye tracking and dynamic audiovisual texts," in *Eye Tracking South Africa*, 2013, pp. 62–66.

[40] P. Kiefer, I. Giannopoulos, A. Duchowski, and M. Raubal, "Measuring cognitive load for map tasks through pupil diameter," in *Geographic Information Science*, 2016, pp. 323–337.

[41] E. H. Hess and J. M. Polt, "Pupil size in relation to mental

activity during simple problem-solving," *Science*, vol. 143, no. 3611, pp. 1190–1192, 1964.
[42] P. Ekman, W. V. Friesen, M. O'Sullivan, and K. Scherer, "Relative importance of face, body, and speech in judgments of personality and affect." *Journal of personality and social psychology*, vol. 38, no. 2, p. 270, 1980.
[43] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction." in *2003 Conference on computer vision and pattern recognition workshop*, vol. 5. IEEE, 2003, pp. 53–53.
[44] B. McDaniel, S. D'Mello, B. King, P. Chipman, K. Tapp, and A. Graesser, "Facial features for affective state detection in learning environments," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 29, no. 29, 2007.
[45] A. Graesser, B. McDaniel, P. Chipman, A. Witherspoon, S. D'Mello, and B. Gholson, "Detection of emotions during learning with autotutor," in *Proceedings of the 28th annual meetings of the cognitive science society*. Citeseer, 2006, pp. 285–290.
[46] E. Friesen and P. Ekman, "Facial action coding system: a technique for the measurement of facial movement," *Palo Alto*, vol. 3, no. 2, p. 5, 1978.
[47] P. Ekman, W. V. Freisen, and S. Ancoli, "Facial signs of emotional experience." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1125, 1980.
[48] L. F. Barrett, B. Mesquita, and M. Gendron, "Context in emotion perception," *Current Directions in Psychological Science*, vol. 20, no. 5, pp. 286–290, 2011.
[49] M. Kayyal, S. Widen, and J. A. Russell, "Context is more powerful than we think: contextual cues override facial cues even for valence." *Emotion*, vol. 15, no. 3, p. 287, 2015.
[50] R. Zhi, M. Liu, and D. Zhang, "A comprehensive survey on automatic facial action unit analysis," *The Visual Computer*, vol. 36, no. 5, pp. 1067–1093, 2020.
[51] N. R. Kuncel, M. Credé, and L. L. Thomas, "The validity of self-reported grade point averages, class ranks, and test scores: A meta-analysis and review of the literature," *Review of educational research*, vol. 75, no. 1, pp. 63–82, 2005.
[52] K. Yim, D. D. Garcia, and S. Ahn, "Computer science illustrated: Engaging visual aids for computer science education," in *Proceedings of the 41st ACM technical symposium on computer science education*, 2010, pp. 465–469.
[53] Tobii Pro AB, "Tobii pro lab," Computer software, Danderyd, Stockholm, 2014. [Online]. Available: http://www.tobiipro.com/
[54] J. R. Shapiro and S. L. Neuberg, "From stereotype threat to stereotype threats: Implications of a multi-threat framework for causes, moderators, mediators, consequences, and interventions," *Personality and Social Psychology Review*, vol. 11, no. 2, pp. 107–130, 2007.
[55] S. J. Spencer, C. M. Steele, and D. M. Quinn, "Stereotype threat and women's math performance," *Journal of experimental social psychology*, vol. 35, no. 1, pp. 4–28, 1999.
[56] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "OpenFace 2.0: Facial behavior analysis toolkit," in *Automatic Face Gesture Recognition*, 2018, pp. 59–66.
[57] A. Olsen, "The Tobii I-VT fixation filter," *Tobii Technology*, vol. 21, pp. 4–19, 2012.
[58] D. D. Salvucci and J. H. Goldberg, "Identifying fixations and saccades in eye-tracking protocols," in *Eye tracking research & applications*, 2000, pp. 71–78.
[59] R. J. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," *Mind*, vol. 2, no. 3, p. 4, 2003.
[60] T. Busjahn, R. Bednarik, A. Begel, M. Crosby, J. H. Paterson, C. Schulte, B. Sharif, and S. Tamm, "Eye movements in code reading: Relaxing the linear order," in *International Conference on Program Comprehension*, 2015, pp. 255–265.
[61] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *J. of the Royal statistical society: series B*, vol. 57, no. 1, pp. 289–300, 1995.
[62] M. Endres, W. Weimer, and A. Kamil, "An analysis of iterative and recursive problem performance," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 2021, pp. 321–327.
[63] I. Polycarpou, "Computer science students' difficulties with proofs by induction: an exploratory study," in *Proceedings of the 44th annual Southeast regional conference*, 2006, pp. 601–606.
[64] L. d. Moura and N. Bjørner, "Z3: An efficient SMT solver," in *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2008, pp. 337–340.
[65] J. M. Schumann, *Automated theorem proving in software engineering*. Springer Science & Business Media, 2001.
[66] K. A. Gamage, R. G. Pradeep, and E. K. de Silva, "Rethinking assessment: The future of examinations in higher education," *Sustainability*, vol. 14, no. 6, p. 3552, 2022.
[67] L. F. Barrett, "Was Darwin wrong about emotional expressions?" *Current Directions in Psychological Science*, vol. 20, no. 6, pp. 400–406, 2011.
[68] R. Netzel, B. Ohlhausen, K. Kurzhals, R. Woods, M. Burch, and D. Weiskopf, "User performance and reading strategies for metro maps: An eye tracking study," *Spatial Cognition & Computation*, vol. 17, no. 1-2, pp. 39–64, 2017.
[69] M. Hegarty, R. E. Mayer, and C. E. Green, "Comprehension of arithmetic word problems: Evidence from students' eye fixations." *Journal of educational psychology*, vol. 84, no. 1, p. 76, 1992.
[70] A. Susac, A. Bubic, M. Planinic, M. Movre, and M. Palmovic, "Role of diagrams in problem solving: An evaluation of eye-tracking parameters as a measure of visual attention," *Physical Review Physics Education Research*, vol. 15, no. 1, p. 013101, 2019.
[71] M. Manoharan and B. Kaur, "Mathematics teachers' perceptions of diagrams," *International Journal of Science and Mathematics Education*, pp. 1–23, 2022.
[72] C. Buckley and C. Nerantzi, "Effective use of visual representation in research and teaching within higher education," *International Journal of Management and Applied Research*, vol. 7, no. 3, pp. 196–214, 2020.
[73] D. Yoon and N. H. Narayanan, "Mental imagery in problem solving: An eye tracking study," in *Proceedings of the 2004 symposium on Eye tracking research & applications*, 2004, pp. 77–84.
[74] M. S. Hussain, O. AlZoubi, R. A. Calvo, and S. K. D'Mello, "Affect detection from multichannel physiology during learning sessions with autotutor," in *International conference on artificial intelligence in education*. Springer, 2011, pp. 131–138.
[75] S. Craig, S. D'Mello, B. Gholson, A. Witherspoon, J. Sullins, and A. Graesser, "Emotions during learning: The first steps toward an affect sensitive intelligent tutoring system," in *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*. Association for the Advancement of Computing in Education (AACE), 2004, pp. 264–268.
[76] H. A. Elfenbein and N. Ambady, "On the universality and cultural specificity of emotion recognition: a meta-analysis." *Psychological bulletin*, vol. 128, no. 2, p. 203, 2002.