

# Relating Reading, Visualization, and Coding for New Programmers: A Neuroimaging Study

Madeline Endres  
Computer Science and Engineering  
University of Michigan  
endremad@umich.edu

Zachary Karas  
Department of Psychology  
University of Michigan  
zackar@umich.edu

Xiaosu Hu  
Department of Psychology  
University of Michigan  
xiaosuhu@umich.edu

Ioulia Kovelman  
Department of Psychology  
University of Michigan  
kovelman@umich.edu

Westley Weimer  
Computer Science and Engineering  
University of Michigan  
weimerw@umich.edu

**Abstract**—Understanding how novices reason about coding at a neurological level has implications for training the next generation of software engineers. In recent years, medical imaging has been increasingly employed to investigate patterns of neural activity associated with coding activity. However, such studies have focused on advanced undergraduates and professionals. In a human study of 31 participants, we use functional near-infrared spectroscopy to measure the neural activity associated with introductory programming. In a controlled, contrast-based experiment, we relate brain activity when coding to that of reading natural language or mentally rotating objects (a spatial visualization task). Our primary result is that all three tasks—coding, prose reading, and mental rotation—are mentally distinct for novices. However, while those tasks are neurally distinct, we find more significant differences between prose and coding than between mental rotation and coding. Intriguingly, we generally find more activation in areas of the brain associated with spatial ability and task difficulty for novice coding compared to that reported in studies with more expert developers. Finally, in an exploratory analysis, we also find a neural activation pattern predictive of programming performance 11 weeks later. While preliminary, these findings both expand on previous results (e.g., relating expertise to a similarity between coding and prose reading) and also provide a new understanding of the cognitive processes underlying novice programming.

## I. INTRODUCTION

In recent years, the software engineering community has increasingly used medical neuroimaging to understand the cognitive processes behind programming [1]–[7]. Unlike eye-tracking or other biometric methods, neuroimaging pinpoints brain regions activated while completing specified tasks. Many of the neuroimaging studies in software engineering have compared programming to reading or spatial manipulation, two skills with well-understood cognitive structures. As broadly summarized in Table I, such studies have generally found striking similarities between code comprehension and prose reading [1]–[3], and one study has found similarities between spatial reasoning and data structure manipulation [6].

Neuroimaging studies of programmers have the potential to improve our understanding of expertise, to inform software engineering pedagogy, and to guide tool development and

retraining (see Floyd *et al.* [3, Sec. II-D] for a summary). Critically, however, to the best of our knowledge all of the software engineering neuroimaging studies thus far have only studied programming *experts* that are either professionals or students with multiple years of experience (e.g., [2, Sec. 3.3]). Tantalizingly, one study [3] found that coding became more neurologically similar to reading for programmers with even greater expertise. However, to realize the potential of neuroimaging for understanding software engineering expertise, we must also directly observe true *novices*. Studying novice programmers is critical for exploring how cognitive processes for coding evolve. In this paper, we seek to close this gap by presenting the first neuroimaging study of novice programmers during code comprehension (cf. [2], [3]).

Studying novice programmers presents several challenges relative to studying experts. First, we must create experimental stimuli that are amenable for novices with little to no previous coding experience; the coding stimuli in previous neuroimaging studies all involve constructs (e.g., trees [6]) or software engineering tasks (e.g., code review [3]) unfamiliar to novices. Second, we must pay particular care to recruit participants with equivalent programming expertise; even in an introductory course, some students have substantially more programming experience than others. Finally, we propose and use an experimental protocol that follows up with participants months later to assess their programming progression using a written assessment; to our knowledge, no previous neuroimaging studies in software engineering have involved time-delayed outcome measurements.

To better understand the cognitive processes of novice programmers, we use functional near-infrared spectroscopy (fNIRS) to conduct a controlled neuroimaging study with 31 participants with no previous programming experience, all enrolled in the same introductory computing class. We conduct these scans during the first third of the class. We compare participants’ brain activation patterns while coding to those while reading prose or using spatial reasoning (i.e., mentally rotating 3D objects). We also use a written assessment at the

Experiment	Like reading?	Like spatial reasoning?	Novices?
Siegmund <i>et al.</i> (2014) [1]	✓		?
Siegmund <i>et al.</i> (2017) [2]	✓		?
Floyd <i>et al.</i> (2017) [3]	✓		?
Huang <i>et al.</i> (2019) [6]		✓	?

TABLE I: “What is coding like in programmers’ brains?” We use this informal summary of selected previous work to motivate and contextualize the experiments in this paper.

end of the semester to conduct a preliminary exploration of an aspect of learning, assessing if novices’ brain activation patterns while coding can predict their future programming ability.

We find that, for novices, coding is a working memory intensive task that is neurally distinct from both a spatial task and prose reading ( $p < 0.01$ ,  $q < 0.05$ ). Unlike previous work with experts which generally reports strong similarities between coding and reading [1]–[3], we observe more significant and substantial differences between coding and reading than we do for coding and spatial tasks. This indicates that novices rely heavily on visiospatial cognitive processes while coding. Finally, we observe that particular activation patterns at the beginning of a course can predict how well students perform on a final programming assessment 11 weeks later; in general, the *less* similar the activation patterns are between coding and mental rotation, the *better* an individual performs ( $r = 0.48$ ,  $p = 0.006$ ). This may indicate that novices who use more problem-solving intensive strategies at the beginning of the semester (i.e., find programming more challenging) make less progress. We also compare our results to those of previous neuroimaging studies with more expert software engineers, and we close with a discussion of the implications of our results on introductory programming pedagogy and future software engineering research.

## II. BACKGROUND

We give an overview of material necessary for understanding the methods and experiments in this paper. In Section II-A, we discuss neuroimaging and fNIRS, and in Section II-B, we present relevant notation. In Section II-C, we discuss the cognition behind spatial ability, and in Section II-D, we discuss the cognition behind reading. See Section IX for a discussion of work more directly related to software engineering.

### A. Neuroimaging and fNIRS

Brain activity and cognitive processes can be studied using *functional neuroimaging* techniques. We focus on functional near infrared spectroscopy (fNIRS). It is non-invasive, avoids the ionizing radiation present in other methods (e.g., PET, CT), and can measure activity in brain regions not accessible to some invasive techniques (e.g., electrocorticography). Importantly, fNIRS offers higher spatial resolution than EEG, and higher temporal resolution than fMRI, which is important for studies relating a brain region’s contribution to a specific task. Finally, fNIRS can be used in more natural and ecologically-valid environments (e.g., standard desktop computer use, etc.)

compared to alternatives like fMRI (which requires participants to lie still in a small tube and also complicates the use of keyboards [7]). These properties motivate our decision to use fNIRS for a study of novice software engineers.

fNIRS makes use of the *hemodynamic response*, or change in neuronal blood flow to active brain regions, to measure brain activity [8]. fNIRS measures this via the use of near-infrared light: transmitters and receivers are placed on a “cap” worn by participants. Oxygen-rich and oxygen-poor blood have different light absorption properties, so the hemodynamic responses in a given brain region between a transmitter and receiver pair (referred to as a *channel*) can be measured over time. fNIRS measures concentration changes in such oxygenated and deoxygenated blood. The number of fNIRS publications had doubled every 3.5 years since 1992 [9], and it has been used to study human development, injury, and psychiatric conditions [9]–[12].

For our purposes, the use of fNIRS imposes two key experimental constraints: contrast-based design and task duration. First, fNIRS experiments typically involve participants completing tasks (e.g., mentally rotating objects or solving programming problems) while time-series data is recorded. Carefully controlled experiments are necessary, in which the activity observed during one task is contrasted against the activity observed during another. This allows confounding brain activations (e.g., motor cortex activity from moving the lungs to breathe) to be eliminated from consideration.

Second, because fNIRS is based on the hemodynamic response, care must be taken to model [13], [14] the onset of neuronal blood flow (which peaks slightly after stimuli are presented [15], [16]) and the design must avoid saturation and weaker signals for tasks involving long activity [17]. As a result, the hemodynamic response can typically be studied only in experiments with brief stimuli (e.g., under 30 seconds per question). Furthermore, fNIRS is only able to penetrate a few centimeters down into the brain. More specifically, the near-infrared light can reach a depth that is roughly half the distance between a transmitter-receiver pair, depending on the wavelengths and light intensities used [18].

Despite these limitations, fNIRS specifically, and medical imaging in general, are growing in popularity for use in software engineering studies (e.g., [1]–[7], [19]–[22]). They provide a physically-grounded insight into cognition without relying on potentially unreliable self-reporting [6].

### B. Neuroscience Vocabulary and Notation

**Vocabulary:** The cerebrum of the human brain is composed of two (largely symmetric) hemispheres, left and right, and four primary lobes: frontal, temporal, parietal, and occipital. Loosely, the *frontal lobe* is at the front of each hemisphere, the *temporal lobe* is on the side of each hemisphere, the *parietal lobe* is at the top of each hemisphere, and the *occipital lobe* is at the back of each hemisphere. Activation is called *bilateral* if both hemispheres are activated and *lateralized* if one hemisphere activates disproportionately. Throughout this paper, we will use various schema to refer to locations on the

cerebrum’s *cortex*, the brain’s outer layer of neural tissue. One such schema is *Brodmann Areas*, an anatomical classification system for the cortex [23]. Brodmann areas (BAs) divide the cortex into 52 bilateral regions based on architectural neurological features. Many BAs also have associated neurological functions. For instance, BAs 41 and 42 are associated with auditory processing. We will also sometimes refer to regions by their common names or location on a lobe. Three important regions mentioned in this paper are Wernicke’s area (left hemisphere, back of the parietal lobe), Broca’s area (left hemisphere, lower frontal lobe), and the dorsolateral prefrontal cortex or DLPFC (bilateral in the frontal lobe). Wernicke’s and Broca’s areas are strongly associated with language functions while the DLPFC is associated with working memory.

**Notation:** In this paper, we will use the neuroimaging notation  $Task A > Task B$  to indicate the contrast between brain activation patterns for two experimental tasks. The results of these contrasts are reported as statistical  $t$ -values corresponding to each fNIRS channel. These  $t$ -values range from  $-8$  to  $+8$ ; a positive  $t$ -value indicates that the specified brain area was *more* active during Task A than Task B, while a negative  $t$ -value indicates that the specified brain area was *less* active during Task A than Task B. Values closer to  $-8$  and  $+8$  represent stronger activation contrasts between the two tasks, and only areas with significant contrast ( $p < 0.01$ ) after correction for multiple comparisons ( $q < 0.05$ ) are reported. Finally, we note that contrast tests are directional: a significant contrast in  $Task A > Task B$  does not imply that the inverse contrast  $Task B > Task A$  will also produce significant results. This is because the differences in the inverse contrast may be too small to be statistically significant.

### C. Spatial Reasoning and Cognition

We now turn to a discussion of Spatial Reasoning and its neurological representations. *Spatial Reasoning* refers to an individual’s general ability to mentally manipulate objects and encompasses skills such as mental rotation, mental folding, pattern recognition, and spatial perception [24]. Spatial reasoning has been shown to correlate with performance in a variety of activities including mathematics [25], [26], general engineering [27], and programming [28], [29]. Spatial ability is also malleable and can be improved through training [30].

Part of spatial reasoning, mental rotation involves the imagined rotation of a two- or three-dimensional object around an axis in three dimensional space [31]. Figure 2a depicts one of the mental rotation stimuli we used (see Section III-C). The difficulty of a mental rotation problem is determined by the size of the angle of rotation; Shepard and Metzler found that the time a participant took to solve a given problem increased linearly with respect to the angle of rotation between the two objects [32]. In this paper, we use mental rotation ability as a validated proxy for more general spatial reasoning ability.

Generally, neuroimaging has found that mental rotation activates the posterior parietal and occipital cortices (BAs, 7, 17, 19, 39, and 40—see Zacks [31] for a survey). While bilateral,

the parietal and occipital activation tends to be slightly stronger in the right hemisphere; the right parietal lobe in particular is believed to be important for spatial ability and spatial attention tasks [33]. Many mental rotation neuroimaging studies have also revealed bilateral activation in the supplementary motor cortex, an area associated with motor control and planning (BA 6) [31]. This frontal activation is most common for mental rotation tasks that allow motor stimulation strategies.

### D. Reading and Cognition

Next, we include a brief summary of the neurological processes associated with reading. Neuroimaging has revealed that language is supported by a complex network of cognitive areas generally lateralized to the left hemisphere (see Price [34] and Vigneau *et al.* [35] for surveys). Some language processes are localized to specific structures in the brain, while other language processes arise from a distributed network of areas with multiple functions [34].

Two key left-hemisphere brain areas associated with reading are Broca’s area and Wernicke’s area. Located in the frontal lobe, Broca’s area (BAs 44 and 45) plays an important role in language production and, to a lesser extent, language comprehension [35]. Wernicke’s area is in the posterior temporal lobe and is associated primarily with comprehension of both spoken and written language [35].

## III. EXPERIMENTAL SETUP AND DESIGN

We now present our experimental design for understanding the neurological basis for novice programmers.<sup>1</sup> Our experiment was conducted in two parts: an initial fNIRS scan and a written followup assessment. During the fNIRS scan, participants were shown three types of stimuli: code comprehension, mental rotation, and prose reading. During the written post-test, participants completed a validated language-agnostic programming assessment. All participants were enrolled in the same 15-week introductory programming course. The initial fNIRS scans were held during the first third of the semester while the written post-test was held during the last week of the semester. This design allows the controlled exploration of the relationships and contrasts between reading, coding, and spatial reasoning for novice programmers. It also opens a preliminary investigation of neurological factors that might be predictive for programming. In the rest of this section we describe our recruitment protocol (subsection III-A); outline our fNIRS data collection, experimental setup, and stimuli (subsections III-B and III-C); and describe our followup programming assessment (subsection III-D).

### A. Participant Recruitment

Participants were all recruited from the same CS1 course at the University of Michigan, a large public US university, via a combination of email, forum posts, and an in-class presentation. To be eligible, participants had to be over 18, have no prior programming experience, only be enrolled in

<sup>1</sup>A replication package containing all of our recruitment and stimuli materials can be found at [https://github.com/CelloCorgi/ICSE\\_fNIRS2021](https://github.com/CelloCorgi/ICSE_fNIRS2021).

that one programming course, and have been available to attend the study.

Recruitment occurred during the third week of the fifteen-week semester, and the fNIRS scans were carried out in the third through fifth week of the semester. All participants were thus in the first third of their first programming course while completing the scan. Participants received \$20 in compensation. In total, we collected fNIRS data from 37 participants. Data from 31 participants passed our analysis quality threshold (see Section IV). The final 31 participants (24 female, 7 male) ranged in age from 18 to 21.

Beyond the initial fNIRS scan, we also followed up with participants via email at the end of the semester, inviting them to attend an additional written programming assessment. Of our 31 participants, 23 participated in this written post-test. Post-test participants were compensated an additional \$20.

During participant selection, we were keenly aware that confirming the absence of previous programming ability for a diverse population is a challenging task. To mitigate self-selection bias, we implemented checks for participant programming ability in three places: recruitment, pre-screening, and statistical validation of scores on a written programming test. During recruitment, both in-person and written, we emphasized that participants could have no prior programming experience of any kind. Prospective participants were explicitly told, in person, that even minimal practice or exposure to textual or visual programming languages (such as Scratch [36] or MIT App Inventor) counted as prior experience.

During prescreening, participants indicated if they “had any prior programming experience” with one of “Yes”, “No”, or “Other/unsure”. We only retained participants who selected “No” outright. The presence of the “Other/unsure” option mitigates some self-selection bias in experience reporting. We also asked potential participants to indicate concurrent and previous course enrollment from a list of courses at our university. Courses that contained “programming” or “code” in their syllabus or description precluded study participation. These questions eliminated 18% of pre-screening respondents.

Finally, at the same time as the demographics questionnaire, participants were given a brief programming test. The test consisted of twelve pseudocode multiple-choice questions and is a validated measure of CS1 concepts [37]. We expected low scores: novices should have no prior programming exposure (beyond the first few weeks of their current course). Indeed, we found an average score of 23% (random guessing yields 20%). There was no statistically significant difference between participant scores and a random distribution. We believe that these three mitigations help account for self-selection bias issues and give confidence that our pool contains novices (but acknowledge that the issue is reduced, rather than eliminated).

We also observe that unusually for software engineering studies, a majority of our population (77%) were female. We believe that the high ratio of females in our population is caused by a combination of: a more gender-balanced population pool than most software engineering studies, fNIRS data signal, and self selection bias. First, the course we recruited

from is fairly gender balanced: around 45–50% of students identify as women, compared to 22% for CS overall at our university. Second, by chance, two-thirds of the participants who we were unable to analyze due to fNIRS signal quality (i.e., measurement noise) were male. Third, women often volunteer for college studies at higher rates than men and for different reasons [38] while men are more likely to have some prior programming experience [39] and thus be excluded from our study.

### B. fNIRS Data Collection and Setup

Each participant’s fNIRS data was collected during a single session which lasted 1.5 hours. First, participants gave informed consent and filled out a short demographic survey. Next, participants watched a training video preparing them for the scan, a 30–45 minute process that involved fitting each participant for the fNIRS cap by moving hair to optimize sensor contact with the scalp, and thus, signal quality.

During the fNIRS scan, the participant sat in a chair facing a monitor wearing the fNIRS cap. The room was kept dim to reduce the amount of ambient light that could interfere with the fNIRS data collection. Participants were also instructed to stay as still as possible. Each participant was shown 90 stimuli: 30 mental rotation stimuli, 30 reading-based stimuli, and 30 coding-based stimuli. All stimuli asked the participant to choose one of two answers. Participants indicated their answer by pressing a corresponding key on a standard keyboard. For each stimuli, participants had up to 30 seconds to respond. The 90 stimuli were in randomized order and were broken into three blocks of 30 questions, each containing 10 stimuli of each type. Between stimuli, participants were shown a fixation cross for 2–10 seconds. Between blocks, participants had an optional longer break to rest and/or drink water.

We now present technical information about our fNIRS device and cap. We used a CW6 fNIRS system (TECHEN, Milford, MA, USA) with 690 nm and 830 nm wavelengths. It has fiber-optic cables which transmit light from the device to sensors connected to the participant’s cap. These sensors are either transmitters that emit light or receivers that detect light. As a result, fNIRS collects a participant’s hemodynamic response only along a set of pre-defined channels between transmitters and receivers. The location, number, and coverage of these channels is determined by the cap design. We used a probe configuration similar to that used in Huang *et al.* [40] with dense coverage of transmitter-receiver pairs in the occipital, frontotemporal, and frontal regions. In total, 15 receiver and 30 transmitter fibers were used, yielding 55 channels from which data were collected, covering Brodmann areas 6–9, 17–19, 21, 22, 39–41, and 44–47. The data from these channels were then analyzed using both NIRS Brain AnalyzIR Toolbox [41] and custom scripts written in MATLAB. A picture of our cap coverage is in Figure 1; it includes areas identified in previous mental rotation studies as well as key language areas.

Our number of channels is higher compared to many previously published fNIRS papers in software engineering,



Fig. 1: fNIRS cap used in our experiments. Red circles are light transmitters while blue circles are light receivers. fNIRS is only able to observe brain activation on channels between nearby transmitters and receivers.

giving broader coverage [4], [19]. We used two cap sizes to accommodate different head circumferences (58 cm and 60 cm). Signals were sampled at 50 Hz.

### C. fNIRS Stimuli

We now turn to a description of the content of our fNIRS stimuli. As mentioned in Section III-B, participants were shown three categories of stimuli: mental rotation, reading-based stimuli, and coding-based stimuli. All three types asked the participant to choose between two answers *A* and *B*.

We use mental rotation stimuli adapted from Peters and Battista’s Mental Rotation Stimulus Library [42]. These stimuli are designed to induce brain activation associated with mentally rotating 3D shapes, one facet of visiospatial cognition (See section II-C). In each mental rotation stimulus, the participant was asked to choose which of two objects was a possible rotation of a third object (see Figure 2a for an example). To admit direct comparison with previous work, our mental rotation stimuli are the same stimuli used by Huang *et al.* when analyzing data structure manipulation brain activation with experienced software developers [6].

For the reading stimuli, we use sentence completion tasks adapted from official *Graduate Record Examination* practice exam questions [43], an assessment required for admission to many graduate programs. For each stimulus, participants were asked to read a sentence and choose the appropriate word or phrase to fill a blank (see Figure 2b for an example).

For the coding stimuli, we created a corpus of short code snippets that use constructs familiar to introductory computing students. Specifically, they contained boolean logic, while loops, for loops, and arrays. These are all early core concepts in most introductory curricula [44] and are also covered in our institution’s CS1. Unfortunately, we were not able to directly reuse coding stimuli from previous neuroimaging studies as they are generally geared toward expert programmers and thus contain constructs unfamiliar to novices. For the array-based questions, however, we were able to adapt stimuli created by Huang *et al.* [6]. For each coding stimulus, participants were asked to choose either the correct output or return value of a short code snippet (see Figure 2c for an example).

### D. Followup Programming Assessment

At the end of the semester (10–12 weeks after the fNIRS scans), we invited participants to complete a written programming test. Along with the programming test, participants also completed a battery of cognitive, and behavioral assessments. For the programming assessment, we used the *Second CS1 Assessment* (SCS1), a validated language-agnostic measure of CS1 programming ability [37]. The SCS1 contains 27 multiple choice questions and takes one hour. It covers Boolean logic, while loops, for loops, arrays, if statements, functions, and recursion. There are three types of questions: definition questions, code tracing questions, and code replacement questions. Due to COVID-19, we were unable to hold the test in person. Rather, participants completed an online version of the SCS1 over a proctored video call. Responses were then checked for timing-based anomalies to ensure participants did not rush through the test.

## IV. ANALYSIS METHODOLOGY

We now go over our methodology for analyzing the fNIRS data. Broadly, there are three stages in our analysis pipeline: preprocessing, individual modeling, and group level modeling.

1) *Preprocessing*: The raw data, in the form of light intensity values, were converted into optical density data by calculating the fluctuations in light absorption by the presence of either oxygenated (HbO) or deoxygenated (HbR) blood. The optical density data were then converted into an HbO/HbR signal using the Modified Beer-Lambert law. We ran a general linear model (GLM) with pre-whitening and robust least squares to fit the data [45].

2) *Individual Subject Modeling*: After the hemodynamic response was modeled for each subject, quality control checks were implemented to limit the amount of noise in the group-level model. The signal-to-noise ratio, anticorrelation of HbO and HbR, and brain-activation plots were considered when deciding whether to exclude individual blocks or whole participants. The signal-to-noise ratio was calculated as a ratio between the absolute signal mean and standard deviation, with a threshold set at 0.9. As a result, 16 blocks were excluded from further analysis (see Section III-B for a discussion of our experimental block setup). In an ideal hemodynamic response function, the levels of HbO will increase as the levels of HbR decrease and vice-versa [46], so the correlation between these two levels should be negative. Thirteen additional blocks that did not show this pattern were excluded from further analysis.

Next, the brain activations estimated by the GLM for *All Conditions*  $>$  *Rest* were plotted onto brain models using a photogrammetry-based localization method [47], at which point the loci of activity could be examined and scrutinized. We expected activity in the visual cortex (as this is a visual experiment), as well as in the inferior frontal gyrus during the reading task (as this is a well-documented region for language processing). Thirty seven blocks that did not pass this activation pattern check were excluded. These criteria were not mutually exclusive. In total, 59 blocks from 31 participants were included in the group-level modeling.

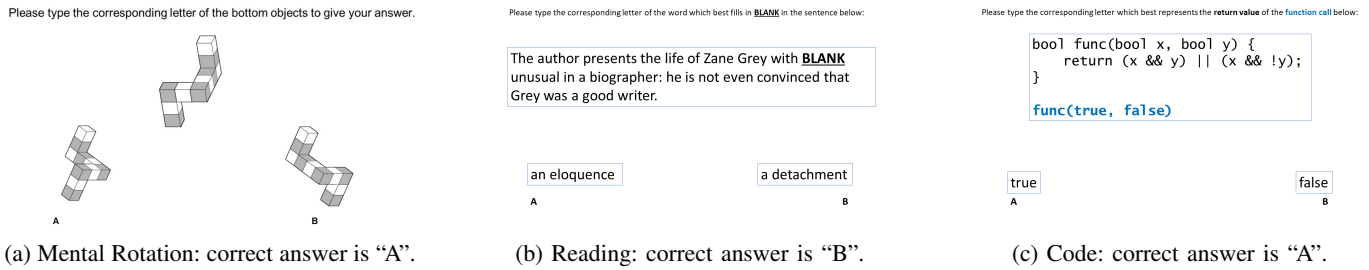


Fig. 2: Example fNIRS Stimuli for mental rotation (a), reading (b), and coding (c) tasks.

3) *Group Modeling*: We used a linear mixed effects model for group level analysis, contrasting *Task* > *Baseline* activations to estimate task-related brain activations and brain-behavior correlations. Lastly, we applied a false-discovery rate (FDR) threshold correction ( $q < 0.05$ ) to account for the multiple-comparison issue.

## V. VALIDATION

In this section we validate that the brain activation patterns we observe during mental rotation and reading align with those established by previous work. Specifically, we present our results for the contrasts *Mental Rotation* > *Rest* and *Reading* > *Rest*. We provide brain activation visualizations for *Mental Rotation* > *Rest* and *Reading* > *Rest* in Figures 3 and 4 respectively. We also provide a tabular view of our results with specific  $t$ -values in Table II (see Section II-B for an overview of  $t$ -values and  $A > B$  notation).

In *Mental Rotation* > *Rest*, we observe significant bilateral activation in the occipital and parietal lobes (BAs 7, 17, 18, 19, 39). The occipital lobe (BAs 17, 18, 19) is associated with the visual cortex and is responsible for tasks such as image and pattern recognition, both visiospatial processes. The parietal activation (BAs 7, 39) is also localized in regions associated with spatial tasks including spatial reasoning and mental rotation [33], [48].

Our results are therefore consistent with previous mental rotation neuroimaging; in his meta-review of mental rotation imaging studies, Zacks also identified BAs 7, 17, 19, and 39 as active during mental rotation tasks [31]. We do not observe significant supplementary motor cortex activation (BA 6), another area often active during mental rotation tasks. This is not too surprising, however, as the supplementary motor cortex is most strongly activated in tasks which encourage participants to use motor simulation strategies [31]; we did not prompt participants to use such strategies in our experiment.

In *Reading* > *Rest*, we observe significant occipital, parietal, and prefrontal activation lateralized to the left hemisphere, aligning with previous work. We observe significant activation in both Broca’s and Wernicke’s areas, widely considered two of the most important language areas [34]. We also observe significant activation in the left dorsolateral prefrontal cortex (See section II-B, BA 46), a region associated with attention and working memory. Regarding the occipital activation, while we observe some bilateral activation, significant activation is

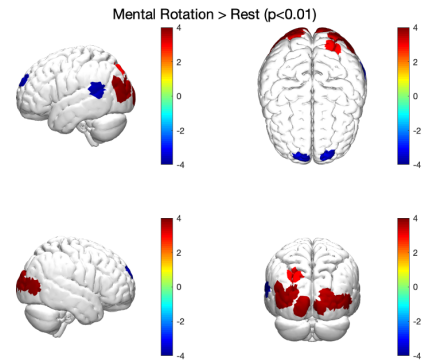


Fig. 3: Baseline Mental Rotation Activation: Red indicates regions more activated during the mental rotation task while blue indicates regions more activated during rest. Note that all significant mental rotation activation is located in the back of the head in the parietal and occipital lobes.

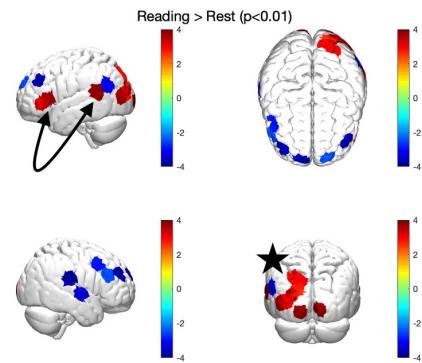


Fig. 4: Baseline Reading Activation: Red indicates regions more activated during the reading task while blue indicates regions more activated during rest. Note the reading activation in Broca’s and Wernicke’s areas (the double arrow) as well as the left-lateralized occipital and parietal activation (the star).

substantially more widespread in the left hemisphere. The left occipital cortex has been found to correspond with word and letter specific pattern recognition [49].

Our activations for rotation and reading align with previous work: significant occipital and parietal activation for rotation, and occipital and prefrontal activation for reading, including Broca’s and Wernicke’s areas. Rotation activation is bilateral while reading is primarily in the left hemisphere. This validation gives confidence for both construct and internal validity (i.e., that our protocol measures what we think it measures and does so correctly).

## VI. EXPERIMENTAL RESULTS

We now present the results of our experiment probing the neurological connections between reading, mental rotation, and programming for novice programmers (See Section II-B for an overview of relevant notation and vocabulary (e.g.  $A > B$ ). We focus our results around three research questions:

- RQ1—Programming Activation: What areas of the brain activate when novice software engineers program?
- RQ2—Comparative Activation: How does the coding brain activation of novices compare to their brain activation during mental rotation and during reading?
- RQ3—Prediction: Are there connections between coding brain activation patterns at the beginning of CS1 and their programming performance at the end of the course?

### A. RQ1—Programming Activation

To determine which areas of the brain activate when novice software engineers program, we present the results of the contrast  $Code > Rest$ . That is, we test which brain areas significantly distinguish programming from a resting state ( $p < 0.01$ ,  $q < 0.05$ ). We also discuss the functionality of the distinguishing brain regions. Figure 5 contains a brain activation visualization for  $Code > Rest$ , and we provide a tabular view of our results with specific  $t$ -values in Table II.

While coding, novices exhibit significant occipital activation (BAs 17, 18, 19). While bilateral, we observe somewhat stronger right hemisphere activity. Functionally, the occipital cortex is associated with visual processing, and it includes areas such as the primary visual cortex (BA 17) and visual association area (BA 18). This occipital activation is the strongest activation we observe in  $Code > Rest$ , with three out of the five channels with  $t$ -values greater than five.

Beyond occipital activation, we also observe significant posterior parietal activation, primarily in the angular gyrus (BA 39). This activation is bilateral: the other two channels with  $t$ -values greater than five both cover BA 39, one in each hemisphere. In the left hemisphere, the angular gyrus is important for language-related tasks [50]. Some researchers also include BA 39 in Wernicke’s area, one of the two main brain cortex regions associated with natural language processing. However, we do not observe activation while coding in the regions most commonly associated with Wernicke’s area: BAs 22 and 40. The angular gyrus is also strongly associated with spatial cognition tasks including spatial orientation (e.g., distinguishing left from right), spatial attention, numerical

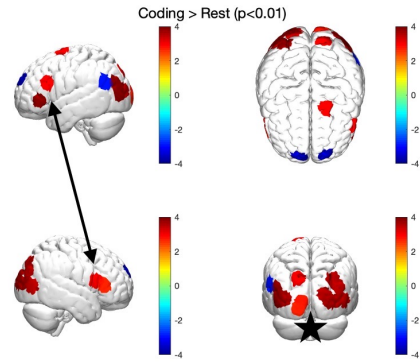


Fig. 5: Significant Coding Activation: Red indicates regions more activated during the coding task while blue indicates regions more activated during rest. Note the widespread bilateral activation in both the DLPFC (the arrows) and the occipital and posterior parietal cortices (the star).

computation, and mental rotation [31], [51]. While the spatial functionality of the angular gyrus is bilateral, many spatial tasks, including mental rotation, are concentrated in the right hemisphere [51]. Therefore, the bilateral activation of the angular gyrus indicates that *novices use both language and spatial cognitive processes while programming*.

We also observe significant activation in the frontal cortex. Specifically, we observe bilateral activation in the DLPFC (BA 46) and activation in the left superior premotor cortex (BA 6). The premotor cortex is associated with motor processing, and it has also been found to activate during visiospatial tasks including mental rotation [31]. The DLPFC is associated with working memory; lower activation in this region corresponds with worse performance on working-memory intensive tasks such as complex problem solving [52]. The significant bilateral DLPFC activation, therefore, indicates that *novices find programming a challenging and working memory intensive task*.

Novices engage brain regions associated with language and spatial cognition, as well as regions associated with increased demand for attention and executive function (via neural activity in  $Code > Rest$ ,  $p < 0.01$ ,  $q < 0.05$ ).

### B. RQ2—Significant Comparative Activation

We now compare novices’ coding brain activation to their activation during mental rotation and reading by presenting our findings for the contrasts  $Code > Mental\ Rotation$  and  $Code > Reading$ . We provide brain activation visualizations for  $Mental\ Rotation > Rest$  and  $Reading > Rest$  in Figures 6 and 7, and we provide a tabular view of our results with specific  $t$ -values in Table II. Our high level finding is that while coding, mental rotation, and reading are all neurally distinct tasks, *we observe more substantial differences between coding and reading than we do between coding and mental rotation*.

Brain Region	<i>Rotation &gt; Rest</i>	<i>Reading &gt; Rest</i>	<i>Code &gt; Rest</i>	<i>Code &gt; Rotation</i>	<i>Code &gt; Reading</i>
<b>Frontal Cortex</b>					
Left DLPFC (BA 46)		3.32 – 3.32	3.30 – 3.95	2.78 – 2.96	3.76 – 3.76
Right DLPFC (BA 46)		-2.62 – -2.62	2.78 – 3.27	3.79 – 3.79	2.99 – 4.72
Broca’s Area (Left BAs 44 and 45)		3.76 – 3.76			-3.54 – -3.54
IFG (Right BAs 44 and 45)			2.78 – 3.27	3.11 – 3.79	2.99 – 2.99
Left Supplementary Motor Cortex (BA 6)			3.45 – 3.45	3.45 – 3.45	3.22 – 5.32
Right Premotor Cortex (BA 6)		-2.98 – -2.98			
Left BA 8		-4.07 – -4.07	3.30 – 3.30	2.96 – 2.96	-4.93 – 3.93
Right BA 8		-3.95 – -2.62			2.67 – 4.72
Left BA 9	-6.26 – -6.26	-4.07 – -2.63	-7.49 – -7.49		-5.10 – 3.93
Right BA 9	-7.46 – -7.46	-6.69 – -2.67	-6.13 – -6.13		2.67 – 4.72
<b>Temporal Cortex</b>					
Wernicke’s Area (Left BAs 22, 40)	-6.44 – -6.44	5.91 – 5.91		6.51 – 6.51	-4.77 – -4.77
Right BA 21					2.79 – 2.79
Right BA 22		-3.65 – -3.65			3.15 – 3.15
Right Auditory Cortex (BA 41)		-5.81 – -5.81			6.70 – 6.70
<b>Parietal Cortex</b>					
Left BA 7	3.38 – 3.38	3.15 – 3.17	3.64 – 3.64	2.60 – 2.60	
Left Angular Gyrus (BA 39)	4.95 – 4.95	-3.19 – -3.19	-2.99 – 5.28		6.86 – 6.86
Right Angular Gyrus (BA 39)	6.83 – 6.83		3.68 – 5.15	-3.49 – 3.98	4.29 – 4.29
<b>Occipital Cortex</b>					
Left BA 17	5.78 – 6.36	8.11 – 8.11	2.75 – 2.83	-3.26 – -3.23	-4.48 – 4.78
Right BA 17	4.17 – 4.17	4.29 – 4.29		-2.84 – -2.84	-2.91 – -2.91
Left BA 18	4.69 – 6.36	3.06 – 8.11	2.75 – 3.84	-3.23 – -3.23	-4.48 – -4.48
Right BA 18	4.18 – 6.48	4.29 – 4.29	5.95 – 5.91	-2.84 – -2.84	-2.91 – 5.26
Left BA 19	4.52 – 7.76	2.71 – 4.16	3.84 – 5.46		
Right BA 19	3.95 – 6.83		3.68 – 5.95	-3.49 – -3.49	3.54 – 5.65

TABLE II:  $t$ -value statistics for *Mental Rotation > Rest*, *Reading > Rest*, *Code > Rest*, *Code > Mental Rotation*, and *Code > Reading*. All reported results are significant ( $p < 0.01$ ) and pass our false discovery threshold ( $q < 0.05$ ). Blank cells indicate no significant effect was found in that region. Results closer to +8 or -8 indicate more significant activation or deactivation. We highlight results with  $t$ -values less than -5 or greater than +5.

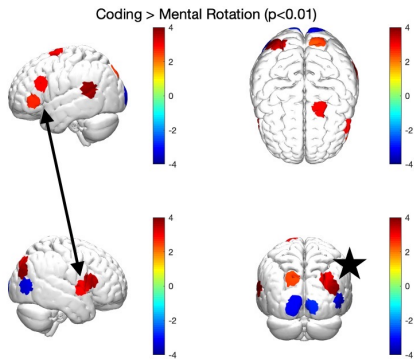


Fig. 6: Activation Contrast between Programming and Mental Rotation: Red indicates regions more activated during coding while blue indicates regions more activated during mental rotation. Note that compared to mental rotation, coding has stronger bilateral frontal activation (the arrow) and right posterior parietal activation in the angular gyrus (the star).

For *Coding > Mental Rotation*, we observe several significant differences in activation. First, when coding, participants exhibit more bilateral frontal activation than while mentally rotating objects. This comparative activation is in the DLPFC and the premotor cortex (BAs 6 and 46), areas commonly as-

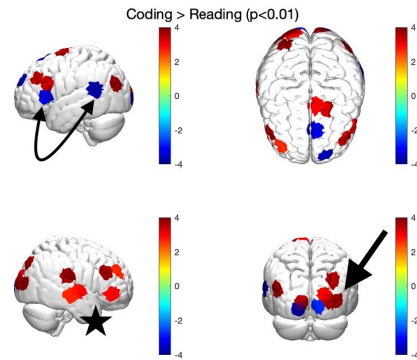


Fig. 7: Activation Contrast between Programming and Reading: Red indicates regions more activated during coding while blue indicates regions more activated during reading. Note that reading has comparatively more activation in Broca’s and Wernicke’s areas (the double arrow), while coding has substantially more right frontal activation (the star) and more right-lateralized occipital / parietal activation (the single arrow).

sociated with working memory and spatial manipulation [31], [52]. Similarly, we observe comparatively high coding activation in the right angular gyrus, a region also connected with spatial reasoning [51]. We find it intriguing that many of the



areas with more activity for coding than for mental rotation are associated with spatial reasoning, ostensibly the process measured by the mental rotation stimuli. This may imply that coding is a challenging task that is actually *more* spatially intensive for novices than simple mental rotation.

When considering a contrast  $A > B$ , it is important to distinguish between a comparison that is positive because  $A$  is greater than  $B$  (called a “true activation”) vs. one that is positive only because  $B$  is negative (a “strong deactivation”). All of the significant *Coding > Mental Rotation* differences discussed so far are channels with either positive activation or no significant activation in *Rotation > Rest*. Thus, they are all true differences between coding and mental rotation. On the other hand, while we also observe a very strong comparative activation ( $t = 6.51$ ) in *Coding > Mental Rotation* in Wernicke’s area, this is a facet of the same channel’s strong deactivation in *Rotation > Rest* instead of a true activation.

For *Coding > Reading*, we also observe significant differences in activation: coding has comparatively stronger activation throughout the right hemisphere and in the premotor cortex while reading has stronger activation in Broca’s and Wernicke’s areas. All of the significant differences in the left hemisphere including Broca’s and Wernicke’s areas are true contrasts; that is, none of them are caused by a significant deactivation in one of the *Task > Rest* comparisons. In the right hemisphere, several of the apparent activations for coding are caused by strong deactivation in *Reading > Rest*. Even so, we observe true comparative right-hemisphere activations: the right occipital, angular gyrus, and DLPFC are all more activated while coding than while reading.

While we observe that Coding is neurologically distinct from both mental rotation and reading, we also observe more substantial differences in *Coding > Reading* than in *Coding > Mental Rotation*. In *Coding > Reading*, there are four channels with  $t$ -values greater than  $+5$  or less than  $-5$ , while in *Coding > Mental Rotation*, there is only one (see Table II). Furthermore, the strong channel difference in *Coding > Mental Rotation* is not particularly compelling as it is caused by deactivation. Taken together with the previous functional analysis, this trend hints that for novices, coding is a more spatially based and a less language-based cognitive process.

We find that, for novices, coding is neurally distinct from both reading and spatial reasoning. Coding engages regions associated with working memory more than does either reading or rotation, indicating that programming is a more cognitively challenging task. However, we observe more significant substantial differences in *Coding > Reading* than we do in *Coding > Rotation*: novices may rely heavily on spatial reasoning while coding.

### C. RQ3—Prediction

We now turn to an exploratory analysis of connections between observed brain activation patterns and participants’

final programming assessment scores. To do so, we use *Representational Similarity Analysis* (RSA), a common Psychology approach, to correlate brain activity interactions with scores on the programming post-test (see Kriegeskorte *et al.* [53] for an introduction to RSA). We test if the brain activation similarity between mental rotation and coding (*Mental Rotation × Coding*) or the brain activation similarity between reading and coding (*Reading × Coding*) are correlated with SCS1 scores. We calculate these correlations for the right hemisphere, left hemisphere, right hemisphere frontal, left hemisphere frontal, and occipital regions for a total of five statistical tests per hypothesis and ten total test–hypothesis pairs.

We find a correlation that remains significant after applying the Bonferroni correction for multiple comparisons [54]. Specifically, we find a significant medium negative correlation between (*Mental Rotation × Coding*) in the right frontal region and programming post-test score ( $r = -0.48$ ,  $p = 0.0006$ , adjusted Bonferroni  $p = 0.006$ ): the *less similar* the neural activation patterns for coding and rotation, the *better* the final programming assessment outcome.

Notice that coding also elicited stronger right frontal activation than the mental rotation task. It is therefore possible that the more individuals engage the right frontal, in a way that is dissimilar from its engagement during spatial processing, the less progress they make over the course of the semester. The hypothesis is similar to the observation that the more dissimilar is the activity between novice readers and their language, the less progress they make in learning to read [55]. In the Psychology literature, one common approach for testing such a hypothesis would be to investigate correlations between significant channel activations and response time. However, while in some settings response time can be used as a proxy for difficulty, our coding stimuli were not designed with that consideration; we do not find a statistically significant correlation ( $r = -0.095$ ,  $p = 0.4866$ ) and lack enough information to either substantiate or refute that hypothesis.

In an exploratory analysis relating initial brain activation patterns and post-test programming scores, we find that *less similar* patterns of activation for coding and mental rotation in the right frontal hemisphere at the start of the semester predict better outcomes on the end-of-semester final programming assessment ( $r = -0.482$ ,  $p = 0.006$ ).

## VII. DISCUSSION

In this section, we discuss the implications of our experimental results. In particular, we consider how the coding brain activation patterns of novice programmers compare to those of more experienced software developers (Section VII-A) and also discuss future research directions (Section VII-B).

### A. Novices vs. Experts

In this section, we discuss how our results compare to those observed in neuroimaging studies of expert developers. Generally, we observe more right-hemisphere activation, more

engagement of visiospatial processes, and less engagement of language processes than is seen in experts. For example, all significant activation areas observed by Siegmund *et al.* were in the left hemisphere, a majority coinciding with established language regions [1]. Similarly, Floyd *et al.* found that for experts, programming becomes increasingly less distinguishable from reading, a left-lateralized cognitive activity. In this context, our work helps establish an experience-based lateralization shift: novice programmers generally exhibit bilateral activation, especially in regions associated with visiospatial processing, while expert developers see increasingly left-lateralized activation centered in language-associated regions.

However, not all extant studies of experts observe a strong connection between reading and programming: in their study examining *code writing* (as opposed code reading, the focus of other neuroimaging studies including our own), Krueger *et al.* observed significantly more right-brain activation in spatial areas during code writing than prose writing [7]. More investigation is needed to see if code writing exhibits a similar lateralization trend with expertise. However, it is possible that code writing consistently remains a more spatial activity.

### B. Future Directions and Implications

To the best of our knowledge, this is the first paper to use medical imaging to explicitly investigate novice programmer coding brain activation patterns and their correlates. As a result, beyond replications and meta-analyses, which are more common in Psychology (e.g., [56]) but not yet as prevalent in Computer Science, many of the research implications relate to building on the baselines established by the results presented here. We focus on two dimensions: how programmers learn other computing activities at a cognitive level, and how learning programming in general compares to established neurological theories for learning other disciplines.

For the former, we note that recent medical imaging research on programming has focused on program comprehension and code review, with a lesser emphasis on data structures and code writing [1]–[3], [6], [7]. Other activities remain unexplored. To take one example, it is unknown whether boolean logic has a significant spatial cognitive component. While general logic has been studied, particular paradigms, such as circuit design, may be processed differently by humans, potentially suggesting alternate training or tool-support approaches. Other experimental protocol paradigms are also relatively unexplored: almost all software engineering neuroimaging research consists of showing fixed, static stimuli. Even relatively foundational experimental structures in Psychology, such as priming, masking, and recall are unexplored. For example, building on the influential work of Chase and Simon [57], Psychologists have studied the relationship between chess expertise and the ability to recall or reason about briefly-presented random chess boards [58]. While Siegmund *et al.* randomized aspects of programs to study comprehension [2], using such paradigms to tease apart programming expertise neurologically remains unexplored.

For the latter, we observe that a number of theories and hypotheses about how humans learn various subjects, from second languages [59] to musical instruments [60], have been posited in the literature. We believe that it will be fruitful to investigate whether a sequential model or a more spatial encoding strategy (see Margulieux [24]) best describes learning to program. Based on our results, our preliminary speculation is that spatial encoding is indeed a key general strategy employed by novices that may decrease in importance over time. If true, this would have implications for the use of domain-specific strategies in skills-based training. A more concrete investigation for programming is merited.

## VIII. LIMITATIONS AND THREATS TO VALIDITY

Although our experiments and analysis provide significant evidence about novice programmers and spatial ability, our results may not generalize. We consider a number of threats to validity and discuss how our approach mitigates them.

We note that fNIRS experiments are dependent on transmitters and receivers for infrared light (see Section II-A): if no pairs are present for a relevant portion of the brain, activity there cannot be measured. We mitigate this potential source of false negatives in two ways. First, adapt a validated cap design proposed by Huang *et al.* for use in software engineering and spatial ability studies [6]. Second, information from other medical imaging approaches, such as fMRI, which do not depend on transmitter placement, is used to determine which brain regions to measure [1], [3], [7].

We also consider issues of construct validity: are we measuring what we claim to be measuring (e.g., spatial ability, introductory programming, etc.)? While there are multiple aspects to spatial ability, we use mental rotation, an established paradigm for investigating spatial ability, both in psychology in general [33], [48], [61] and in computer science in particular [6], [62]. For introductory programming, we make use of the SCS1, a validated assessment [37].

Finally, all of our subjects are students at the same large US university. This aspect of participation selection may limit the generality of our results to other populations.

## IX. RELATED WORK

We place our results in context with respect to three broad categories of previous work.

*a) Spatial Skills and Programming:* There is a positive relationship between programming and spatial ability and [28], [29], [63], [64]. Parkinson and Cutts found that “spatial skills typically increase as the level of academic achievement in computer science increases” [29]. Furthermore, Parker *et al.* found that spatial reasoning is better mediating variable for affluence discrepancies in computer science than computing access [64]. There have also recently been studies establishing a causal transfer between spatial reasoning training and computer science performance. Cooper *et al.* and Bockmon *et al.* ran studies with high school and university programmers, finding that those who participated in additional spatial training

performed better on a final programming test [62], [65]. Margulieux’s spatial encoding strategy (SpES) framework relates the cogitative processes behind spatial ability and learning to program [24]. SpES hypothesizes that strong spatial reasoning ability helps novice programmers use general strategies for mentally encoding non-verbal information.

Our results provide context and nuance to such claims: we find that novice programmers *do* use spatial cognitive processes while programming (RQ1) and that the degree of dissimilarity between patterns of neural activity for coding and spatial tasks *can* predict final outcomes (RQ3).

b) *Reading and Programming*: From documentation to code review to requirements elicitation to code summarization, many software engineering activities involve a significant reading component [66]–[68]. Experimentally, several studies report a correlation between overall programming ability and the ability to read a program and describe its function in natural language [69], [70] or posit natural language reading as a basis for code comprehension [71], [72]. Some models extend to training: Fedorenko *et al.* hypothesize that “pedagogies for developing linguistic fluency” can inform how to train programmers, based on a perceived similarity between learning programming and second language learning [73]. Their hypothesis was supported by a recent study by Prat *et al.* which found that natural language aptitude was a significant factor in predicting programming success [72].

Our results elaborate on how such claims apply to novice programmers: while novices *do* use language cognitive processes when coding (RQ1), we find that coding and reading are more *different* for novices than are coding and spatial ability, suggesting they rely more on spatial reasoning when coding.

c) *Medical Imaging and Software Engineering*: Following the pioneering work of Siegmund *et al.* [1], a number of papers have used medical imaging techniques to investigate software engineering activities (e.g., [1]–[5], [7], [19]–[21]). Related to our work in particular, Yu *et al.* used fNIRS to compare mental rotation tasks to data structure manipulation [6].

A key distinction of our work is that those studies focus on programmers with years of experience. Explicit investigations of programming expertise using neuroimaging are relatively rare, and tend to involve either proxies such as undergraduate grades [3] or comparisons between graduates or professionals and undergraduates (e.g., Siegmund *et al.* measure 8 students and 3 professionals [2, Sec. 3.3]). While Floyd *et al.* found that coding and prose tasks are more similar in terms of neural activity for senior undergraduate than for mid-level undergraduates [3] (i.e., as programmers become more experienced), our results provide evidence that *the pattern continues*: as programmers become less experienced, programming and reading show less cognitive similarity (RQ1, RQ2).

## X. CONCLUSION

Neurological understandings of how novices engineer software has implications for training, pedagogy and tool development. In a study of 31 participants, we use fNIRS to compare the neural activity for introductory programming,

reading and spatial reasoning tasks in a controlled, contrast-based experiment. We find that **all three tasks — coding, prose reading, and mental rotation — are mentally distinct for novices**. This clarifies previous findings that they may be more similar in experts [3] or for complex data structures [40]. However, while those tasks are neurally distinct, we find **more significant and substantial differences between prose and coding than between mental rotation and coding**. Intriguingly, we find generally more activation in areas of the brain associated with spatial ability and task difficulty while coding compared to that reported studies with more expert developers. Finally, in an exploratory analysis, we find that **certain patterns of neural activity at the start of the semester are predictive of end-of-semester outcomes**, opening the door for future experiments to model such phenomena more directly. To the best of our knowledge, this is the first study to focus specifically on novice programmers and to make use of a significant time-delayed outcomes assessment. While preliminary, these findings both elaborate on previous results (e.g., relating expertise to a similarity between coding and prose reading) and also provide a new understanding the cognitive processes underlying novice programming.

## ACKNOWLEDGEMENTS

We acknowledge the partial support of the NSF (CCF 1908633, CCF 1763674) as well as both the Center for Research on Learning and Teaching and also the Center for Academic Innovation at the University of Michigan. Additionally, we thank Jessica Kim for her help understanding the fNIRS setup, and we thank Yu Huang for sharing the fNIRS cap from her previous work. Finally, we thank our undergraduate research assistants Anne Fitzpatrick, Annie Li, and Serena Chan for their logistical help and their help piloting fNIRS stimuli.

## REFERENCES

- [1] J. Siegmund, C. Kästner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann, “Understanding understanding source code with functional magnetic resonance imaging,” in *Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 378–389.
- [2] J. Siegmund, N. Peitek, C. Parnin, S. Apel, J. Hofmeister, C. Kästner, A. Begel, A. Bethmann, and A. Brechmann, “Measuring Neural Efficiency of Program Comprehension,” in *Foundations of Software Engineering*, 2017, pp. 140–150. [Online]. Available: <http://doi.acm.org/10.1145/3106237.3106268>
- [3] B. Floyd, T. Santander, and W. Weimer, “Decoding the representation of code in the brain: An fmri study of code review and expertise,” in *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. IEEE, 2017, pp. 175–186.
- [4] T. Nakagawa, Y. Kamei, H. Uwano, A. Monden, K. Matsumoto, and D. M. German, “Quantifying programmers’ mental workload during program comprehension based on cerebral blood flow measurement: a controlled experiment,” in *Companion proceedings of the 36th international conference on software engineering*, 2014, pp. 448–451.
- [5] S. Fakhoury, Y. Ma, V. Arnaudova, and O. Adesope, “The effect of poor source code lexicon and readability on developers’ cognitive load,” in *International Conference on Program Comprehension*, 2018.
- [6] 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. [Online]. Available: <https://doi.org/10.1109/ICSE.2019.00053>

- [7] 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*, 2020.
- [8] R. B. Buxton, K. Uludağ, D. J. Dubowitz, and T. T. Liu, "Modeling the hemodynamic response to brain activation," *Neuroimage*, vol. 23, pp. S220–S233, 2004.
- [9] D. A. Boas, C. E. Elwell, M. Ferrari, and G. Taga, "Twenty years of functional near-infrared spectroscopy: introduction for the special issue," 2014.
- [10] S. Lloyd-Fox, A. Blasi, and C. Elwell, "Illuminating the developing brain: the past, present and future of functional near infrared spectroscopy," *Neuroscience & Biobehavioral Reviews*, vol. 34, no. 3, pp. 269–284, 2010.
- [11] A.-C. Ehlis, S. Schneider, T. Dresler, and A. J. Fallgatter, "Application of functional near-infrared spectroscopy in psychiatry," *Neuroimage*, vol. 85, pp. 478–488, 2014.
- [12] H. Obrig, "NIRS in clinical neurology — a 'promising' tool?" *Neuroimage*, vol. 85, pp. 535–546, 2014.
- [13] Scicurious, "IgNobel prize in neuroscience: The dead salmon study," *Scientific American Blog Network*, Sep 2012. [Online]. Available: <https://blogs.scientificamerican.com/scicurious-brain/ignobel-prize-in-neuroscience-the-dead-salmon-study/>
- [14] C. M. Bennett, M. Miller, and G. Wolford, "Neural correlates of interspecies perspective taking in the post-mortem atlantic salmon: an argument for multiple comparisons correction," *Neuroimage*, vol. 47, no. Suppl 1, p. S125, 2009.
- [15] R. N. Henson, C. J. Price, M. D. Rugg, R. Turner, and K. J. Friston, "Detecting latency differences in event-related bold responses: application to words versus nonwords and initial versus repeated face presentations," *Neuroimage*, vol. 15, no. 1, pp. 83–97, 2002.
- [16] R. Aamand, T. Dalsgaard, Y.-C. Lynn Ho, A. Moller, A. Roepstorff, and T. Lund, "A NO way to BOLD?: Dietary nitrate alters the hemodynamic response to visual stimulation," *NeuroImage*, vol. 83, 07 2013.
- [17] M. A. Lindquist, J. M. Loh, L. Y. Atlas, and T. D. Wager, "Modeling the hemodynamic response function in fMRI: efficiency, bias and mis-modeling," *Neuroimage*, vol. 45, no. 1, pp. S187–S198, 2009.
- [18] M. Ferrari and V. Quaresima, "A brief review on the history of human functional near-infrared spectroscopy (fnirs) development and fields of application," *Neuroimage*, vol. 63, no. 2, pp. 921–935, 2012.
- [19] Y. Icutani and H. Uwano, "Brain activity measurement during program comprehension with nirs," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. IEEE, 2014, pp. 1–6.
- [20] J. Duraes, H. Madeira, J. Castelhana, C. Duarte, and M. C. Branco, "WAP: Understanding the Brain at Software Debugging," in *International Symposium on Software Reliability Engineering*, 2016, pp. 87–92.
- [21] J. Castelhana, I. C. Duarte, C. Ferreira, J. Duraes, H. Madeira, and M. Castelo-Branco, "The Role of the Insula in Intuitive Expert Bug Detection in Computer Code: An fMRI Study," *Brain Imaging and Behavior*, May 2018.
- [22] N. Peitek, J. Siegmund, C. Parnin, S. Apel, J. Hofmeister, and A. Brechmann, "Simultaneous Measurement of Program Comprehension with fMRI and Eye Tracking: A Case Study," in *Symposium on Empirical Software Engineering and Measurement*, 2018, to appear.
- [23] K. Brodmann, *Brodmann's: Localisation in the cerebral cortex*. Springer Science & Business Media, 2007.
- [24] L. E. Margulieux, "Spatial encoding strategy theory: The relationship between spatial skill and stem achievement," in *Proceedings of the 2019 ACM Conference on International Computing Education Research*, ser. ICER '19, 2019, p. 81–90.
- [25] M. Hegarty and M. Kozhevnikov, "Types of visual-spatial representations and mathematical problem solving," *Journal of educational psychology*, vol. 91, no. 4, p. 684, 1999.
- [26] J. Wai, D. Lubinski, and C. P. Benbow, "Spatial ability for stem domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance," *Journal of Educational Psychology*, vol. 101, no. 4, p. 817, 2009.
- [27] S. Sorby, N. Veurink, and S. Streiner, "Does spatial skills instruction improve stem outcomes? the answer is 'yes'," *Learning and Individual Differences*, vol. 67, pp. 209–222, 2018.
- [28] S. Jones and G. Burnett, "Spatial ability and learning to program," *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*, 2008.
- [29] J. Parkinson and Q. Cutts, "Investigating the relationship between spatial skills and computer science," in *Proceedings of the 2018 ACM Conference on International Computing Education Research*, 2018, pp. 106–114.
- [30] D. H. Uttal, N. G. Meadow, E. Tipton, L. L. Hand, A. R. Alden, C. Warren, and N. S. Newcombe, "The malleability of spatial skills: A meta-analysis of training studies," *Psychological bulletin*, vol. 139, no. 2, p. 352, 2013.
- [31] J. M. Zacks, "Neuroimaging studies of mental rotation: a meta-analysis and review," *Journal of cognitive neuroscience*, vol. 20, no. 1, pp. 1–19, 2008.
- [32] R. N. Shepard and J. Metzler, "Mental rotation of three-dimensional objects," *Science*, vol. 171, no. 3972, pp. 701–703, 1971.
- [33] J. C. Culham and N. G. Kanwisher, "Neuroimaging of cognitive functions in human parietal cortex," *Current opinion in neurobiology*, vol. 11, no. 2, pp. 157–163, 2001.
- [34] C. J. Price, "A review and synthesis of the first 20 years of pet and fmri studies of heard speech, spoken language and reading," *Neuroimage*, vol. 62, no. 2, pp. 816–847, 2012.
- [35] M. Vigneau, V. Beaucousin, P.-Y. Herve, H. Duffau, F. Crivello, O. Houde, B. Mazoyer, and N. Tzourio-Mazoyer, "Meta-analyzing left hemisphere language areas: phonology, semantics, and sentence processing," *Neuroimage*, vol. 30, no. 4, pp. 1414–1432, 2006.
- [36] J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond, "The scratch programming language and environment," *ACM Trans. Comput. Educ.*, vol. 10, no. 4, Nov. 2010. [Online]. Available: <https://doi.org/10.1145/1868358.1868363>
- [37] M. C. Parker, M. Guzdial, and S. Engleman, "Replication, validation, and use of a language independent cs1 knowledge assessment," in *Proceedings of the 2016 ACM Conference on International Computing Education Research*, ser. ICER '16, 2016, p. 93–101.
- [38] L. Lobato, J. M. Bethony, F. B. Pereira, S. L. Grahek, D. Diemert, and M. F. Gazzinelli, "Impact of gender on the decision to participate in a clinical trial: a cross-sectional study," *BMC Public Health*, vol. 14, no. 1, pp. 1–9, 2014.
- [39] M. G. Sackrowitz and A. P. Parelius, "An unlevel playing field: Women in the introductory computer science courses," *ACM SIGCSE Bulletin*, vol. 28, no. 1, pp. 37–41, 1996.
- [40] 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 (ICSE)*, 2019.
- [41] H. Santosa, X. Zhai, F. Fishburn, and T. Huppert, "The NIRS brain AnalyzIR toolbox," *Algorithms*, vol. 11, no. 5, May 2018.
- [42] M. Peters and C. Battista, "Applications of mental rotation figures of the shepard and metzler type and description of a mental rotation stimulus library," *Brain and cognition*, vol. 66, no. 3, pp. 260–264, 2008.
- [43] ETS.org. (2020) Gre home. ETS. [Online]. Available: <https://www.ets.org/gre>
- [44] A. E. Tew and M. Guzdial, "Developing a validated assessment of fundamental cs1 concepts," in *Proceedings of the 41st ACM technical symposium on Computer science education*, 2010, pp. 97–101.
- [45] J. W. Barker, A. Aarabi, and T. J. Huppert, "Autoregressive model based algorithm for correcting motion and serially correlated errors in fnirs," *Biomedical optics express*, vol. 4, no. 8, pp. 1366–1379, 2013.
- [46] X. Cui, S. Bray, and A. Reiss, "Functional near infrared spectroscopy (NIRS) signal improvement based on negative correlation between oxygenated and deoxygenated hemoglobin dynamics," *Neuroimage*, vol. 49, no. 4, pp. 3039–3046, 2010.
- [47] X.-S. Hu, N. Wagley, A. T. Rioboo, A. F. DaSilva, and I. Kovelman, "Photogrammetry-based stereoscopic optode registration method for functional near-infrared spectroscopy," *Journal of Biomedical Optics*, vol. 25.
- [48] M. S. Cohen, S. M. Kosslyn, H. C. Breiter, G. J. DiGirolamo, W. L. Thompson, A. Anderson, S. Bookheimer, B. R. Rosen, and J. Belliveau, "Changes in cortical activity during mental rotation a mapping study using functional mri," *Brain*, vol. 119, no. 1, pp. 89–100, 1996.
- [49] L. Cohen, O. Martinaud, C. Lemer, S. Lehericy, Y. Samson, M. Obadia, A. Slachevsky, and S. Dehaene, "Visual word recognition in the left and right hemispheres: anatomical and functional correlates of peripheral alexias," *Cerebral cortex*, vol. 13, no. 12, pp. 1313–1333, 2003.
- [50] S. L. Brownsett and R. J. Wise, "The contribution of the parietal lobes to speaking and writing," *Cerebral Cortex*, vol. 20, no. 3, pp. 517–523, 2010.

- [51] M. L. Seghier, "The angular gyrus: multiple functions and multiple subdivisions," *The Neuroscientist*, vol. 19, no. 1, pp. 43–61, 2013.
- [52] A. K. Barbey, M. Koenigs, and J. Grafman, "Dorsolateral prefrontal contributions to human working memory," *cortex*, vol. 49, no. 5, pp. 1195–1205, 2013.
- [53] N. Kriegeskorte, M. Mur, and P. A. Bandettini, "Representational similarity analysis-connecting the branches of systems neuroscience," *Frontiers in systems neuroscience*, vol. 2, p. 4, 2008.
- [54] S.-Y. Chen, Z. Feng, and X. Yi, "A general introduction to adjustment for multiple comparisons," *Journal of thoracic disease*, vol. 9, no. 6, p. 1725, 2017.
- [55] R. A. Marks, I. Kovelman, O. Kepinska, M. Oliver, Z. Xia, S. L. Haft, L. Zekelman, P. Duong, Y. Uchikoshi, R. Hancock, and F. Hoeft, "Spoken language proficiency predicts print-speech convergence in beginning readers," *NeuroImage*, vol. 201, p. 116021, 2019.
- [56] P. R. Ventura Jr, "Identifying predictors of success for an objects-first cs1," *Computer Science Education*, 2005.
- [57] H. A. Simon and W. G. Chase, "Perception in chess," *Cognitive Psychology*, vol. 4, no. 1, pp. 55–81, 1973.
- [58] Y. Gong, K. Ericsson, and J. Moxley, "Recall of briefly presented chess positions and its relation to chess skill," *PLoS ONE*, vol. 10, no. 3, p. e0118756, 2015. [Online]. Available: <https://doi.org/10.1371/journal.pone.0118756>
- [59] I. Kovelman, S. A. Baker, and L. A. Petitto, "Bilingual and monolingual brains compared: a functional magnetic resonance imaging investigation of syntactic processing and a possible "neural signature" of bilingualism," *Journal of Cognitive Neuroscience*, vol. 20, no. 1, pp. 153–169, 2008.
- [60] R. L. Lathrop, "How students learn music: The psychology of music and music education," *Music Educators Journal*, vol. 56, no. 6, pp. 47–145, 1970.
- [61] I. M. Harris, G. F. Egan, C. Sonkkila, H. J. Tochon-Danguy, G. Paxinos, and J. D. Watson, "Selective right parietal lobe activation during mental rotation: a parametric pet study," *Brain*, vol. 123, no. 1, pp. 65–73, 2000.
- [62] R. Bockmon, S. Cooper, W. Koperski, J. Gratch, S. Sorby, and M. Dorodchi, "A cs1 spatial skills intervention and the impact on introductory programming abilities," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 2020, pp. 766–772.
- [63] S. Fincher, A. Robins, B. Baker, I. Box, Q. Cutts, M. de Raadt, P. Haden, J. Hamer, M. Hamilton, R. Lister *et al.*, "Predictors of success in a first programming course," in *Proceedings of the 8th Australasian Computing Education Conference (ACE 2006)*, vol. 52. Australian Computer Society Inc., 2006, pp. 189–196.
- [64] M. C. Parker, A. Solomon, B. Pritchett, D. A. Illingworth, L. E. Marguilieux, and M. Guzdial, "Socioeconomic status and computer science achievement: Spatial ability as a mediating variable in a novel model of understanding," in *Proceedings of the 2018 ACM Conference on International Computing Education Research*, ser. ICER '18, 2018, p. 97–105.
- [65] S. Cooper, K. Wang, M. Israni, and S. Sorby, "Spatial skills training in introductory computing," in *International Computing Education Research*, 2015, pp. 13–20. [Online]. Available: <https://doi.org/10.1145/2787622.2787728>
- [66] X. Xia, L. Bao, D. Lo, Z. Xing, A. E. Hassan, and S. Li, "Measuring program comprehension: A large-scale field study with professionals," *IEEE Transactions on Software Engineering*, vol. 44, no. 10, pp. 951–976, 2017.
- [67] S. Haiduc, J. Aponte, L. Moreno, and A. Marcus, "On the use of automated text summarization techniques for summarizing source code," in *2010 17th Working Conference on Reverse Engineering*. IEEE, 2010, pp. 35–44.
- [68] P. W. McBurney and C. McMillan, "Automatic source code summarization of context for java methods," *IEEE Transactions on Software Engineering*, vol. 42, no. 2, pp. 103–119, 2015.
- [69] L. Murphy, S. Fitzgerald, R. Lister, and R. McCauley, "Ability to explain in plain english linked to proficiency in computer-based programming," in *Proceedings of the ninth annual international conference on International computing education research*, 2012, pp. 111–118.
- [70] M. Lopez, J. Whalley, P. Robbins, and R. Lister, "Relationships between reading, tracing and writing skills in introductory programming," in *Proceedings of the fourth international workshop on computing education research*, 2008, pp. 101–112.
- [71] T. Busjahn, C. Schulte, and A. Busjahn, "Analysis of code reading to gain more insight in program comprehension," in *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*, 2011, pp. 1–9.
- [72] C. S. Prat, T. M. Madhyastha, M. J. Mottarella, and C.-H. Kuo, "Relating natural language aptitude to individual differences in learning programming languages," *Scientific reports*, vol. 10, no. 1, pp. 1–10, 2020.
- [73] E. Fedorenko, A. Ivanova, R. Dhamala, and M. U. Bers, "The language of programming: a cognitive perspective," *Trends in cognitive sciences*, vol. 23, no. 7, pp. 525–528, 2019.