Causal Relationships and Programming Outcomes: A Transcranial Magnetic Stimulation Experiment

Hammad Ahmad
hammada@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Madeline Endres
endremad@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Kaia Newman
kaian@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Priscila Santiesteban
pasanti@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Emma Shedden
emshedde@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

Westley Weimer
weimerw@umich.edu
University of Michigan
Ann Arbor, Michigan, USA

ABSTRACT

Understanding the relationship between cognition and programming outcomes is important: it can inform interventions that help novices become experts faster. Neuroimaging techniques can measure brain activity, but prior studies of programming report only correlations. We present the first causal neurological investigation of the cognition of programming by using *Transcranial Magnetic Stimulation* (TMS). TMS permits temporary and noninvasive disruption of specific brain regions. By disrupting brain regions and then measuring programming outcomes, we discover whether a true causal relationship exists. To the best of our knowledge, this is the first use of TMS to study software engineering.

Where multiple previous studies reported correlations, we find no direct causal relationships between implicated brain regions and programming. Using a protocol that follows TMS best practices and mitigates for biases, we replicate psychology findings that TMS affects spatial tasks. We then find that **neurostimulation can affect programming outcomes**. Multi-level regression analysis shows that TMS stimulation of different regions significantly accounts for 2.2% of the variance in task completion time. Our results have implications for interventions in education and training as well as research into causal cognitive relationships.

KEYWORDS

Neurostimulation, spatial ability, code reading, data structures

ACM Reference Format:
Hammad Ahmad, Madeline Endres, Kaia Newman, Priscila Santiesteban, Emma Shedden, and Westley Weimer. 2024. Causal Relationships and Programming Outcomes: A Transcranial Magnetic Stimulation Experiment. In *2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE '24), April 14–20, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3597503.3639096

ICSE '24, April 14–20, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0217-4/24/04…$15.00
https://doi.org/10.1145/3597503.3639096

Figure 1: High-level experimental architecture: "Does impairing a brain region influence programming outcomes?"

1 INTRODUCTION

Recently, *neuroimaging studies*, which noninvasively measure brain activity, have been used by software engineering researchers to pinpoint the brain regions most correlated with common programming activities such as comprehension [35, 70, 86, 87], code reading and writing [39], debugging [13, 29], and data structure manipulation [48, 56]. These studies, and subsequent work, identified key cognitive processes correlated with software engineering tasks.

For example, data structures are often described spatially (e.g., "balanced", "length", "height", etc.), suggesting a potential relationship between how humans reason spatially and how they reason about data structures. Spatial reasoning (or spatial visualization) refers to the ability to mentally manipulate three dimensional objects. Huang *et al.* confirmed a *correlative* relationship between spatial visualization and data structure manipulation [48]. Neuroimaging results have the potential to improve our understanding of expertise, to inform pedagogy, and to guide tool development and retraining (see Floyd *et al.* [39, Sec. II-D] for a summary).

The Problem. Despite these potential benefits and despite researcher interest, to the best of our knowledge, no prior neuroimaging study in software engineering has confirmed a *causal* relationship between patterns of neural activation and software engineering activities. Specific causal relationships from one variable to another cannot usually be assessed from an observed association between them [1, 47] (cf. "correlation is not causation", confounds, etc.).

Proposed Solution. To scientifically investigate the plausible existence of a causal connection between spatial visualization and programming tasks, we require an approach that limits the effects of any confounding variables: manipulating and influencing brain regions directly. The desired approach should: (1) admit high-confidence causal inference, (2) comprise a noninvasive process, and (3) apply to indicative software engineering tasks.

We propose the first investigation of a causal relationship between spatial reasoning and programming tasks. We use *Transcranial Magnetic Stimulation* (TMS) to noninvasively and directly impede or facilitate visualization-associated regions [88, Sec. 5] of the brain and then analyze the effects on programmer performance. Unlike medical imaging, TMS induces a current within a region of the brain itself, temporarily changing transmembrane potentials and causing neurons to be more or less excitable. Stimulation that interferes with task performance indicates that the affected brain region is necessary for the task (i.e., establishes causality).

Unlike neuroimaging methods, such as functional magnetic resonance imaging (fMRI), that indicate correlations between brain and behavior, TMS can be used to demonstrate causal brain-behavior relations [69, Sec. 3][82, pp. 595–596].

Our Study. We applied TMS to 16 participants, disrupting three regions of their brains to probe causal relationships between software engineering and neural activity. The regions were stimulated on different days via an established TMS protocol (Section 3.3) and state-of-the-art per-subject brain region localization (Section 3.3.1). To the best of our knowledge, this is also the first such study in software engineering to feature multiple treatments and visits, more naturally admitting both within-one-subject and between-multiple-subjects analyses (cf. previous SE neuroimaging replications using different subjects each time [86]). After TMS, subjects completed a randomized set of 180 tasks: code comprehension, data structure manipulation, and mental rotation. Differences in outcomes (e.g., time) give confidence in a causal relationship (see Figure 1).

Experimental Rigor. Care is necessary to avoid bias as we probe causal relationships (cf. "absence of evidence is not evidence of absence"). We *pre-register* hypotheses (mitigating some threats from researcher bias), correct for multiple comparisons (mitigating some threats of false discovery), use special active controls (mitigating some threats of participant response bias [25]), and conduct some analyses with condition labels anonymized (mitigating some threats from researcher bias). In addition, with over 1,600 minutes of neurostimulated performance, our study involves comparable observation to correlative studies (e.g., 1,300 [39], 600 [86], etc.).

Findings and Contributions. Analyzing standard performance measures (e.g., time taken), not medical scans, we find:

- *"Interpreting computing cognition is not simple."* We find no evidence of causal relationships for multiple previously-published correlations (e.g., for code understanding [86, Sec. 5.1], data structures [48, Sec. V.B], code complexity [70, Sec. III.B], or code writing [56, Sec. 5.2]). That is, disrupting a single region does not uniformly impair performance.
- *"Neurostimulation can affect spatial ability."* We replicate prior findings that stimulation of supplementary motor area degrades mental rotation task completion time. This is important both as a replication and also because it gives confidence that we are applying TMS correctly.
- *"Neurostimulation can affect computing outcomes."* We find that TMS treatment condition contributes to time outcome dispersion in experimental observations (participants completing task stimuli). Via multi-level regression analysis, we find that TMS accounts for 2.2% of the variance in task completion times after accounting for learning effect. **This is a particularly exciting result**, since neurostimulation has been used to improve performance in other domains.
- We make our materials (recruiting, stimuli, analysis scripts, and de-identified data) available for replication. We discuss experiences conducting a TMS study for future researchers.

2 BACKGROUND

In this section, we discuss causal inference in software engineering, provide relevant background on transcranial magnetic stimulation as a neurostimualtion technique, and summarize spatial ability.

2.1 Causation in Software Engineering

Understanding causation is important for many reasons, including the potential of misdirected software engineering research if correlation and causation are confused. For example, some early work on the program repair tool GenProg [99] assumed that correlated components (e.g., fitness functions) were important for success and worth improving [37] — only for subsequent work causally testing that supposition by removing those components [75, 98] to find just as much success without them. Similarly, in deep learning, overlap between training and testing datasets can increase perceived performance. However, some early work assumed that correlated components (e.g., model techniques) were primary drivers of success — only for subsequent work to causally test that supposition (e.g., by renaming variables or otherwise avoiding "contamination" [52, Sec. VIII]). Within the intersection of software engineering and neuroscience, a longitudinal study by Endres *et al.* [33] demonstrated the benefits of medical imaging for pedagogy, evaluating a training method based on prior neuroimaging results [34]. However, the training based on spatial visualization actually produced worse results than technical reading [33, Sec. 7.1], a result not in line with prior correlative studies (e.g., [48]). While other fields place a more direct emphasis on reproducing or replicating findings and following correlative analyses with causal ones (cf. some aspects of the replication crisis in psychology [27]), with some notable exceptions (e.g., [85]), software engineering does not yet have a comparable tradition of accepting negative results or replications.

2.2 Transcranial Magnetic Stimulation (TMS)

Transcranial Magnetic Stimulation (TMS) is a safe and noninvasive technique that is well-established for a variety of clinical and scientific use cases. When administered, TMS produces magnetic fields which stimulate (or disrupt) activity in a region of the brain by inducing an electric current in the neurons of this region [10]. Clinically, TMS is used as a treatment for major depressive disorder, smoking cessation, and obsessive-compulsive disorder, among others (see Section 2.3). It is also a well-established research tool:

in the past 10 years, the National Library of Medicine has recorded over 1000 academic papers published each year which investigate the use of TMS. By using this *neurostimulation* technique to disrupt brain activity and subsequently measuring task outcomes on programming tasks, we can examine potential causal relationships.

Compared to other methods, TMS is a time-efficient way to investigate the causal link between neural activity and programming ability. Other medical approaches that affect the brain in specific areas tend to be quite invasive, requiring implanted electrodes, drug treatments, or neurological surgeries. By contrast, non-medical approaches, such as transfer training or pedagogy, are typically studied over a longer period of time (see Section 8). In such longitudinal A/B studies, the participant dropout rate may reach 70%, primarily due to the effort and time required [45]. Our protocol only required about five hours per participant to establish the control and treatment effects, mitigating participant dropout.

Short applications also eliminate variance and potential confounds in long-term studies. In longitudinal studies, it is important to control environmental factors, such as what a participant is learning elsewhere and variance in intra-individual factors (such as mood, energy, etc.). For example, longitudinal studies on CS student retention span weeks or years that can lead to uncontrolled factors, such as the recent pandemic [33, 83]. By contrast, the direct and immediate effects of TMS can be observed in a controlled environment for a specific trial of programming-related questions.

TMS is well-suited to elevate a correlative neural relationship to one that can be suspected to be causal: (1) it is time-efficient and noninvasive, and (2) minimizes confounding variables.

2.3 Current TMS Applications and Successes

This is the first time TMS has been applied to software engineering, but it has been used successfully in a variety of contexts. We highlight advances in aspects such as creativity, memory, language and mathematics that are relevant for software engineering.

Creativity and Memory. Hertenstein *et al.* clarified the neural basis of creativity (broadly construed as the "use of original ideas to accomplish something innovative"), as well as ways to modulate that creativity, based on TMS stimulation of the prefrontal cortex [46]. They found that deactivating the left prefrontal cortex and activating the right prefrontal cortex with transcranial stimulation is associated with increased creativity, whereas doing the opposite (activating the left, deactivating the right) is significantly associated with decreased creativity. Other activities that fall under this definition of creativity, such as the sudden insight gained when solving anagrams, "aesthetic experience", and "divergent thinking", have been effectively studied for their neural correlates via TMS [17, 73, 81]. Moreover, participants in a TMS study on working memory were tasked to remember various numbers and do addition on them: neurostimulation resulted in a 30% accuracy increase [43].

Language. TMS has been applied to explore language processing. Willems *et al.* observed that stimulating the premotor cortex resulted in increased verb processing speed for manual actions [100]. This is particularly interesting because it not only demonstrates that two seemingly disparate activities can be connected by similar activity in the brain, but also shows that stimulation of one area can improve processing in another (cf. transfer training [19, 33]).

Mathematics. TMS has been employed for brain region causal inference for various mathematical tasks. In one study, TMS significantly improved calculation accuracy for mental arithmetic done on three-digit numbers [96]. In another study, TMS was used to investigate neural models of mathematical cognition via the task of mentally considering the prices of items [54].

Health. In the medical domain, TMS is used to treat several neurological conditions. For example, TMS has been applied to word-finding difficulty (anomia), a common problem in early stages of Alzheimer's disease [21]. TMS is also widely used to counteract the effects and recurrence of depression. TMS applied to 301 previously-untreated participants with depression showed a significant reduction in depressive symptoms and twice the likelihood of remission after 6 weeks [68]. TMS has also diminished the effects of PTSD, OCD, Tourette's, and other mental health conditions [61, 63]. Software engineering may involve stress, novel demands, or toxicity [76], high rates of burnout and depression [102], and lower rates of treatment for such mental health issues [59]. Software-specific challenges can impair the productivity of employees (e.g., the "happy-productive thesis" [22]). Mental health may thus be as relevant to programming as language, creativity, or math.

TMS Summary. TMS has successfully influenced tasks requiring complex interactions in many different areas of the brain, including programming-relevant aspects such as creativity, memory, language, and mathematics. Although TMS has not previously been used for software engineering, we propose its use to investigate the relationship between neural activity and programming.

2.4 Spatial Reasoning and Programming

Spatial reasoning is the capacity to understand, remember, and manipulate the orientation of objects in space, including both physical and abstract objects [62]. The particular task of *mental rotation* involves visualizing 2D or 3D objects in the mind and imagining pivoting them [71]. Mental rotation has been shown to be a significant predictor for ability with many different STEM-related disciplines [16]. While spatial reasoning has been studied for decades [11], it has only been recently linked to programming.

Huang *et al.* observed similar patterns of neural activation between spatial reasoning tasks and programming with tree-based data structures [48]. Endres *et al.* found even greater similarity between neural activation for spatial visualization and programming ability in novices [33]. Given that spatial ability could be a predictor for aptitude with a variety of data structures or programming tasks, it is important to establish a causal, rather than correlative, relationship between it and programming ability.

3 EXPERIMENTAL SETUP AND METHODS

We present our study design for investigating the causal link between neural activity and computation via Transcranial Magnetic Stimulation (TMS). Each individual underwent a localizing (fMRI) scan and two to four subsequent TMS sessions, each on a different day. At each TMS session, an experimental condition was applied: stimulation of one of two spatial reasoning-associated regions or stimulation of an active control (leg-associated) region. After treatment, participants were tested on a set of stimuli. This design allows

for a controlled investigation of a potential causal link between spatial reasoning and program comprehension for programmers.

3.1 Participant Recruitment

We recruited 16 participants via a combination of email, course forums, posters, and in-class presentations. Eligible subjects were required to be 18 years or older, right-handed, native English speakers, have normal to corrected vision, and had at least 1.5 years of programming experience. Due to TMS safety policies, participants were also required to pass a medical screening form. Participants whose medical history indicated any neurological risk factors, drugs active in the central nervous system (e.g., antipsychotics, antidepressants, or recreational stimulants), or poor levels of sleep were excluded from the study [79, 97]. We note that the risks associated with TMS are minimal, with only one known case of a seizure [79].

Because individual humans vary slightly in brain anatomy [3], we scanned each individual to produce a personalized localization. We collected 23 brain scans, of which 16 are part of our final analysis (others dropped out or failed later safety screenings). Additionally, data from one participant was removed from the final analyses due to inconsistencies and outlier data points (i.e., response times more than 2 standard deviations away from the mean). Overall, our final analysis considered 16 participants: 8 male and 8 female.

Background and demographic information were collected from 14 out of 16 participants. 6 of our participants reported being undergraduate students, 4 as graduate computer science students, 2 as software engineers, 1 as a non-computing student, and 1 as a non-computing-related professional. All subjects were also screened for basic programming knowledge of C++. Participants who completed the study in full, which consisted of a localizing anatomical scan and three subsequent TMS sessions, received $125.

3.2 Stimuli and Tasks

After each TMS treatment, participants were shown a varied set of 61 stimuli from three tasks: Code Comprehension, Data Structure Manipulation, and Mental Rotation. In total, participants were presented 183 stimuli over three treatment sessions. We selected stimuli that were short and concise to fit well within the 60-minute effect window of the TMS treatment (see Section 3.3.1). Data structure and mental rotation stimuli were acquired from previously-published studies that examined spatial visualization and programming [33, 48] and thus relate to our research questions. Code comprehension stimuli were taken from previous quizzes and exams administered in a data structures and algorithms course at a large public university in the US. Responses to each stimulus were given by selecting one of two answer choices via the '*A*' or '*B*' keys on a standard laptop keyboard. Stimuli were administered via the popular PyschoPy (version 2022.2.5) package. Individual tasks took 15–60s to complete, with 35 minutes to complete all 61 stimuli. We now describe the stimuli in further detail.

3.2.1 Data Structure Manipulation Task. We obtained a total of 89 validated data structure task stimuli from a prior publication reporting a neural correlation with software engineering tasks [48]. Stimuli cover arrays, linked lists, and trees. Each stimulus included a starting data structure, an operation to perform, and two answer choices (Figure 2a). Answers were either numerical values to describe the outcome of an operation or candidate data structures resulting from an operation. The tree tasks include binary search tree (BST) rotation, insertion, and traversal operations.

3.2.2 Mental Rotation Task. We use both the Huang *et al.* [48] and Endres *et al.* [33] spatial skills stimuli. These include Mental Rotation Stimulus Library questions established by Peters and Battista [71] with varying rotational angle difficulty as well as the Revised Purdue Spatial Visualization Test (PSVT:R II) [105]. PSVT:R II is a standard assessment of different facets of spatial ability. Mental rotation tasks asked participants to compare two 3D objects rotated about an axis (Figure 2b). Participants selected the object that matched the starter object, accounting for rotation. Our stimuli include 56 distinct mental rotation tasks.

3.2.3 Code Comprehension Task. Code comprehension tasks were acquired from exams and quizzes for a data structures and algorithms course at the University of Michigan, a large public university. All tasks have previously been used to assess thousands of undergraduate students on their knowledge of data structures. For each stimulus, participants were asked to trace through snippets of C++ code and select one of two answer choices (Figure 2c). Tasks included deducing the values printed or returned by a function, and analyzing the time and memory complexity of the code. A total of 38 distinct code comprehension stimuli were included in our study.

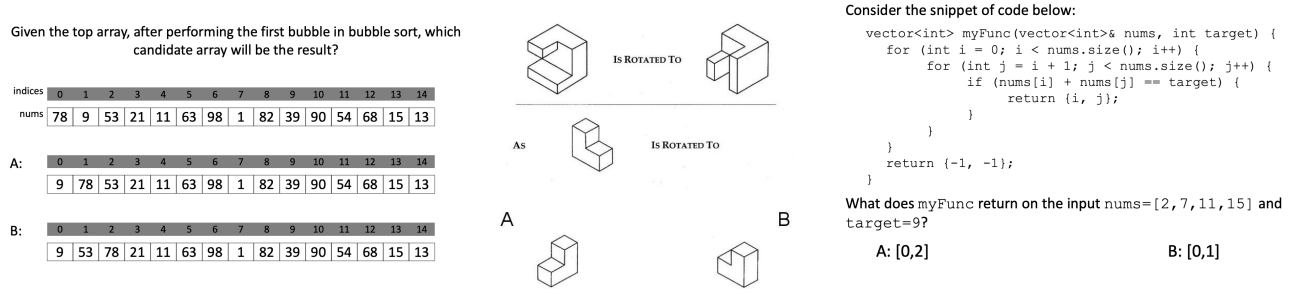
3.3 TMS Treatment

We summarize our experimental design decisions at a high level. We claim no novelty in the mechanics of TMS application — indeed, we intentionally use a high-quality but "off-the-shelf" TMS protocol (see Figure 3) for this application in software engineering. In brief:

(1) "How do we apply TMS at all?" We use a best-practice protocol and off-the-shelf hardware and software (Section 3.3.1).
(2) "How much TMS do we apply?" Following best practices, we find a per-participant stimulation thresholds (Section 3.3.2).
(3) "Where do we apply TMS?" Following best practices, we measure each participant's individual brain anatomy and target brain regions implicated in previous correlative studies (Sections 3.3.3 and 3.3.4).
(4) "How do we minimize bias?" We use a best-practice active control in which an unrelated brain region is stimulated (in a process that still feels like other TMS treatment, Section 3.3.3). We randomize treatment conditions and stimuli and blind conditions when possible (Section 3.3.5).

Knowledge of TMS details (e.g., "theta-burst stimulation") is not necessary to understand our results or their import. TMS can be viewed as an effective"black box" that temporarily impairs brain regions (see Section 2); the remainder of this section provides details relevant for replication and justification of best-practice decisions.

3.3.1 Stimulation Protocol. We applied a continuous theta-burst stimulation (cTBS) protocol consisting of 3 pulses of stimulation at 50 Hz, repeated every 200 ms, for a total of 600 pulses in 40 seconds. The method is an accepted form of stimulation in various psychology and medicine research papers studying TMS effects [49, 93]. This method is effective in providing long-lasting effects of approximately 60 minutes [49]. This is essential for our experiment,



(a) Data structure manipulation stimulus **(b) Mental rotation stimulus** **(c) Code comprehension stimulus**

Figure 2: Example stimuli. Data structures also include linked lists and trees, code also include Big-O complexity.

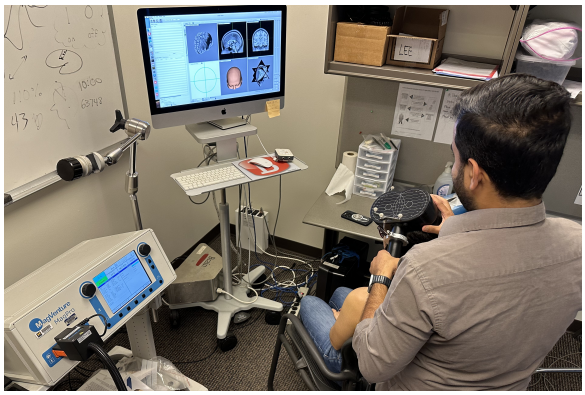


Figure 3: Transcranial magnetic stimulation treatment setup. The researcher (standing right) uses the per-participant localization (top screen) and hand-held magnetic coil (center right) on the scalp of the participant (seated center) to induce current in a brain region.

since effects should last long enough to complete the 35-minute task block presented after TMS treatment, but short enough to limit effects post-study to mitigate safety concerns. The time to complete this protocol is drastically smaller than that of other stimulation protocols (cf. "repetitive TMS" [12]), facilitating recruitment.

We used a well-established stimulation procedure to maximize accuracy and time [49]. cTBS was delivered over the scalp through a MagPro X100 magnetic stimulator and a 90 mm figure-8 coil (MC-B70, MagVenture Inc.). The cTBS protocol was tolerated well by all subjects with no negative side effects reported.

3.3.2 Thresholding. Because each human is slightly different, we use a well-established protocol to determine an appropriate stimulation intensity for each participant. We first find the participant's individualized active motor threshold (AMT) for the first dorsal interosseous muscle (FDI) of the right hand as they contract the FDI [74, 80]. This common method involves stimulating the primary motor cortex on the left hemisphere at various levels with the aim of eliciting a motor evoked potential (MEP) of $\geq 50\ \mu$V peak-to-peak on five out of ten trials while the participant is subjectively contracting the FDI muscle at 20% of maximum. A stimulation threshold that meets such requirements is known as the AMT and allows us to effectively stimulate each participant safely [74, 80]. In most subjects, the lowest stimulation threshold can be found in this manner [80]. To ensure accurate recording of MEPs and AMT, the participant is adjusted with disposable self-adhesive electromyograph (EMG) on their right hand. EMG activity was amplified ($x1000$) with a BioAmp (AD Instruments, USA) using a Powerlab 4/35 system and digitized (10 kHz) and recorded using "Brainsight TMS" neuronavigation software (Rogue Research, Montréal, Canada). Physiological responses were visually monitored because twitches near or around the FDI of the right hand can indicate if stimulation is occurring at the correct positioning [79]. Once AMT was determined for the participant, cTBS stimulations were applied at 80% AMT to comply with commonly-accepted safety standards [79, 97].

3.3.3 Treatment and Control Conditions. Participants were stimulated in multiple brain regions to assess the causal relationship between neural activity and programming. In particular, we stimulated the *primary motor cortex (M1)* (reported as correlated with code understanding [86, Sec. 5.1], data structures [48, Sec. V.B], and code complexity [70, Sec. III.B]) and the *supplementary motor cortex (SMA)* (reported as correlated with code writing [56, Sec. 5.2]). The left primary motor cortex was chosen as the motor sub-area for stimulation since all participants were required to be right-handed.

To ensure that any changes observed in the participant are caused by the stimulation, as opposed to some other general factor (e.g., arousal, attention, altering response to the TMS sounds), we apply an *active control* condition in which the *cranial vertex* (a leg-associated brain region) is stimulated. The vertex region is a commonly-used control in TMS studies with cTBS protocols [28, 60]. Introducing an active control is shown to provide the same sensation of TMS stimulation without affecting the brain areas of interest [51, 65, 78, 89]. An active control thus further mitigates participant response bias [25, 51]. In total, participants were stimulated in three different brain regions, one on each of three different TMS sessions in randomized order.

3.3.4 Stimulation Localization. Every brain is slightly different [3], so we collected individual 3D brain scans to accurately target stimulation on each participant. While some studies report localizing brain anatomy by sight or by feel, we used an fMRI to collect high-resolution imaging following best TMS localization practices [51].

All imaging procedures were conducted on a 3T General Electric MR750 with a 32-channel head coil at the University of Michigan Functional MRI Laboratory. Participants attended a single 45-minute scanning session for brain region localization. High-resolution anatomical images were acquired with a T_1-weighted spoiled gradient recall (SPGR) sequence ($TR = 2300.80$ ms, $TE = 24$ ms, $TI = 975$ ms, $FA = 8^{\circ}$: 208 slices, 1 mm thickness). We obtained estimates of the static magnetic field using spin-echo fieldmap sequences ($TR = 7400$ ms, $TE = 80$ ms; 2.4 mm slice thickness).

Subjects' heads were reconstructed in 3D using the Brainsight TMS neuronavigation software from their T_1 anatomical scans and the locations of the left primary motor cortex (M1), SMA, and cranial vertex were determined for stimulation. The M1 region was identified using axial scans by locating the "hand knob" and hook in MRI images [9, 23, 97]. The SMA region was located by selecting the voxel in individual anatomical scans best corresponding to the Brodmann area definition for pre-SMA and SMA (Talairach coordinates $x = -28, y = 0, z = 48$) [57, 58] (cf. [67]). The cranial vertex control region was located by selecting the intersection of an abscissa between the nasion and the inion, and an abscissa between the left and right tragus on individual structural brain scans [51, 72]. Localization methods used were overseen by two independent TMS experts (not authors on this work), adding confidence.

Localized regions were marked for stimulation as targets via Brainsight's frameless stereotaxy system which uses an infrared camera for monitoring head locations of the participant by tracking reflexive markers attached to the head of the participant [91, Sec. 2]. Head locations are then related to the structural MRI brain data of the participant, guiding precise positioning of the magnetic coil.

3.3.5 Minimizing Bias. In addition to our use of an active control (see Section 3.3.3), we took additional steps to reduce bias. First, participants were not informed of which brain region was being stimulated at the time of the session. Second, participants were not given information on the expected effects [25]. This was single-blind, not double-blind, since the researcher manually targets the TMS coil at the brain region and thus knows the treatment condition. Third, however, after each TMS stimulation was applied, participants were presented with randomized task stimuli on an automated, online platform which required no interaction with the researchers. A final post-test survey was administered on a printed page. We believe that these (non-)interaction procedures help minimize threats associated with participant response bias.

4 ANALYSIS METHODOLOGY

We analyze our results via statistical assessments and modeling. Critically, unlike fMRI-based software engineering papers (which must use nuanced methods to account for large numbers of noisy voxels, etc., when analyzing brain scans, e.g., [39, Sec. IV]), our primary analyses are of the broad form "did the participants in the treatment condition answer the test questions better (or faster) than those in the control condition?". While some modeling sophistication is required (e.g., to account for heterogeneity, see Section 4.2), we never analyze brain scan data.

However, to form robust experimental conclusions, especially involving potential "negative" results, we must minimize the potential for bias, including researcher bias during analysis. In addition to approaches taken in our experimental protocol (e.g., randomization, single-blind, etc., see Section 3) we also follow two practices in our analysis: pre-registering our hypotheses and partial blind analysis.

4.1 Pre-Registration and Bias

Pre-registration is a scientific process in which the "research rationale, hypotheses, design and analytic strategy" are submitted before beginning the study [44]. This helps mitigate biases associated with researchers choosing which results to present post hoc: "pre-registration can prevent or suppress HARKing, p-hacking, and cherry picking since hypotheses and analytical methods have already been declared before experiments are performed" [104]. Similarly, following a discipline of pre-registration may mean that "researchers will not be motivated to engage in practices that increase the likelihood of making a type I error" [44]. While not as common in software engineering (but see the "Registered Reports" track of Mining Software Repositories [85], for example), pre-registration is increasingly adopted by journals and researchers, especially in fields such as psychology and social science (e.g., [90]).

Our hypotheses, such as "TMS stimulation in the SMA or motor cortex will significantly disrupt accuracy and or reaction times on both mental rotation and programming tasks compared to an active control condition (TMS stimulation in the vertex)", were pre-registered with the Open Science Framework (https://osf.io/m4p6e) along with our data collection strategy and statistical analysis methods. This includes our criteria for excluding data and inferring significant correlations.

In addition, our final analysis was conducted blind: labels representing the treatments (vertex, SMA and M1) were randomly coded as `A`, `B`, and `C`, before the analysis strategy was set. This helps mitigate researcher bias in the choice of analysis tools or methods.

Finally, the Benjamini-Hochberg (BH) adjustment was used to correct for multiple comparisons when necessary in evaluating p-values [7]. Prior work has shown the choice of statistical software is important [31]: our analysis primarily used the R package `lme4` and the Python package `scikit-learn`.

4.2 Multi-level Regression Analysis

Our experimental design produces item-level assessment data, where each response to a question contributes an observation to the dataset. We broadly follow the framework of *Item Response Theory* (IRT), a branch of psychometrics which is concerned with the analysis of this type of data [4, 103]. Specifically, we employ multi-level regression models to examine relationships between a response variable, stimulated brain region, and control variables. We linearly model response time and self-reported perceived difficulty, and logistically model accuracy. This is collectively referred to as multi-level regression analysis, or mixed-effects modeling.

We claim no novelty in statistics, and focus our discussion on why this analysis appropriately incorporates important aspects of our data and research hypotheses. For general information about our methods, we refer the reader to [5] and [36, Ch. 8].

4.2.1 Suitability of Multi-level Regression. Multi-level regression analysis is well-suited to handling heterogeneity between groups of observations, such as arise from repeated measures [42]. In our experiment, each participant response (to 150–183 stimuli) is a distinct

data item; these may be correlated due to an underlying person-dependent "skill". We also have repeated measures for each stimulus, as multiple participants answer every question; such observations may be correlated due to variation in question difficulty. We can also posit heterogeneity between content domains (e.g., code comprehension vs. data structures). Such considerations are common in the analysis of item-level assessment data [38, 64]. Multi-level models also perform well with unbalanced group sizes. Our experiment has modest imbalance (e.g., 843 observations of SMA stimulation vs. 939 of M1). Moreover, not all questions have responses from all participants (e.g., from drop-out).

Multi-level regression analysis allows us to test hypotheses about both systematic and heterogeneous TMS effects, as discussed below.

4.2.2 Systematic and Heterogeneous Effects. A mixed-effects model can include independent variables whose effects are systematic (*fixed effects*), heterogeneous (*random effects*), or both. *Interactions* of fixed effects further permit modeling effects that are systematic within specific groups of observations. This is relevant because we hypothesize a systematic TMS treatment effect within each programming task (e.g., data structures vs. code comprehension).

Random effects can pertain to multiple levels of *grouping* in the data. For example, they can model heterogeneity between people, between person-domains, or both. This is relevant because we hypothesize a heterogeneous TMS treatment effect that varies between people, as has been found in TMS studies of other disciplines [10, 17, 46, 54, 73, 74, 80, 81, 95]. That is, some people may improve performance under TMS while others reduce performance.

We are interested in the TMS effect distribution over the population represented by our study subjects. This is mirrored in our experiment design, which features person-specific localization (Section 3.3.4) and person-specific TMS intensity thresholding (Section 3.3.2). Mixed-effects models can express our hypothesized person-dependent TMS effect using a random effect that describes interactions (combinations) of TMS conditions and participants.

4.2.3 Model Specification, Parameter Estimation, and Inference. The dependent variables we consider are per-question accuracy, per-question response time, and perceived difficulty. We first consider plausible effect structures for the available independent variables, based on existing literature and our experimental design [64]. Examples are given in Section 4.2.2 (see replication package for full list). We apply logarithmic transformation to response times to address skew (discussed in Sections 5.1 and 5.3). All models are fit by maximum likelihood estimation (MLE) to the programming and mental rotation data separately. To find the best-fitting candidate model for each dependent variable, we optimize Akaike Information Criterion (AIC), a widely-used model selection metric [2, 14].

We are interested in the TMS treatment condition (i.e., which brain region was stimulated), which may exhibit a fixed or a random effect. If the best-fitting model has a fixed (systematic) TMS effect, we explicitly verify statistical significance via a likelihood ratio "omnibus" test relative to a model without the TMS effect [36, Appendix A.2]. We then pinpoint the source using post-hoc pairwise contrasts, with Benjamini-Hochberg adjustment for the 3 comparisons. Alternatively, if the best-fitting model has a random (heterogeneous) TMS effect, we explicitly verify statistical significance using profile likelihood analysis [5] and parametric bootstrap methods [24] to find the 95% confidence bounds of the statistic.

4.3 Replication

Our replication package contains raw data for de-identified participants as well as relevant analysis information, including scripts, data management, and statistical assumption checking, and is publicly available at https://github.com/hammad-a/ICSE24_TMS.

5 RESULTS

With behavioral and survey data, we ask:

RQ1: Can we replicate prior findings that neurostimulation of the SMA reduces mental rotation completion times?

RQ2: Is there a direct causal relationship between activity in the SMA (or M1) brain region alone and performance?

RQ3: Does neurostimulation of the SMA or M1 brain regions affect objective computing performance outcomes?

RQ4: Does neurostimulation in the SMA or M1 brain region affect self-perceived problem difficulty?

5.1 RQ1: TMS and Mental Rotation

Prior psychology studies using TMS found a causal link between the SMA and mental rotation, but no such link for the M1 [18]. To gain confidence in the accuracy of our results regarding the SMA, M1, and computing, we attempt to replicate this causal link between the SMA and mental rotation.

We find that *TMS stimulation of the SMA impairs response time for spatial reasoning stimuli*, compared to TMS of the vertex region (our active control condition). With $p \leq 0.02$, TMS stimulation of the SMA results in an increase of 0.143 log-seconds in expected per-question log-transformed response time[1] (a 15.3% increase in raw response time, or 1.5 s slower on our median response time of 9.82 s) relative to stimulation of the vertex region.

5.2 RQ2: SMA, M1 and Computing

The *supplementary motor area* (SMA, Broadmann area #6) is a part of the motor cortex that coordinates complex and internally-guided motor actions for extremities. The *primary motor cortex* (M1, Broadmann area #4) is in the anterior bank of the precentral sulcus and is involved in the execution of voluntary, external body movements (such as contracting skeletal muscles).

Overall, we find *no evidence of a causal relationship* between activity in the supplementary motor area and computing outcomes. In particular, we find no question type (data structure, mental rotation, or code comprehension) for which accuracy in the SMA treatment condition and accuracy in the control condition are statistically different ($p \geq 0.81$). Similarly, there is no question type for which response times for the SMA treatment condition and time taken in the control condition are statistically different ($p \geq 0.22$).

We also find *no evidence of a causal relationship* between activity in the primary motor area and computing outcomes for any question type ($p \geq 0.50$ for accuracy, $p \geq 0.73$ for time taken).

Quite surprisingly, **our results do not agree with multiple previously-established correlations.** For instance, for the SMA

[1]We log-transform the dependent variable to address right skew in raw response times (skewness 3.18 → 0.35) and residuals of the optimal-AIC fitted model (2.93 → 0.59).

region, Siegmund *et al.* found a correlation between brain activity and code understanding [86, Sec. 5.1], Huang *et al.* found a correlation between activity and data structure manipulation [48, Sec. V.B], and, most recently, Peitek *et al.* found a correlation between neural activity and comprehension of code with higher complexity metrics [70, Sec. III.B]. Likewise, for the M1 region, Krueger *et al.* found a correlation between it and code writing (as opposed to prose writing) [56, Sec. 5.2]. The lack of evidence of a causal relationship between single-region brain activity and computing outcomes calls into question the research community's understanding of cognition for software engineering tasks. Simply disrupting activity in one region does *not* uniformly result in lower outcomes. Our results suggest that **interpreting cognition for CS is complex**: multiple brain regions could be causally responsible for outcomes for programming tasks (cf. [53]).

Our null results further argue for nuance in pedagogical interventions based on cognition. Indeed, a recent investigation by Endres *et al.* [33] concluded that student training based on spatial visualization produced worse results than technical reading, a result not in line with prior correlative studies (e.g., [48]). The lack of a causal relationship between brain regions associated with spatial reasoning and computing outcomes helps further explain these recent results, and cautions against misdirected software engineering research and pedagogical interventions that may otherwise be undertaken if correlation and causation are confused.

We fit a multi-level linear model to mental rotation responses (see Section 4.2, details in replication package). Critically, our optimal-AIC model contains the TMS condition as a fixed effect. We calculate post-hoc pairwise contrasts between TMS conditions (with Benjamini-Hochberg adjustment), to obtain the significance result ($p \leq 0.02$). This result generalizes over both types of mental rotation stimuli from prior work: we find no significant difference in the impact by stimulus source (see Section 3.2.2).

Of note, we also find no significant difference in response times between TMS stimulation of the M1 and control ($p = 0.18$). Our results thus replicate prior findings that TMS of the SMA impacts mental rotation response times, but that TMS of the M1 does not have a significant effect [18]. **Our replication results give confidence that we have applied TMS correctly.**

5.3 RQ3: TMS and Computing Outcomes

While our analyses for RQ2 find no evidence of a monotonic causal relationship (e.g., "stimulating the SMA alone always reduces performance on data structure questions"), multi-level regression analysis finds that *TMS stimulation does have a statistically significant non-systematic person-dependent effect on response time.* Per Section 4.2, we produce a best-fit model of response times.[2] Table 1 shows point estimates and 95% confidence intervals.

The confidence interval for the standard deviation of the "Participant by Brain Region Stimulated" random effect excludes zero, indicating a significant effect. The estimated proportion of variance explained (PVE, equal to the variance of estimate interest divided by the sum of all variances) of this effect is 2.2%. The 95% confidence interval for that figure is (0.7%, 4.0%), as calculated using the methods in Section 4.2 (see replication package for derivation).

Table 1: Mixed-effects model parameter estimates predicting log-transformed response times. The "C.I." columns give the confidence interval for the standard deviation estimate of the corresponding random effect. Critically, the "Participant by Brain Region" interval (bolded) excludes 0, indicating a statistically significant person-specific effect involving TMS neurostimulation.

Random Effect	Vari.	Std. Dev.	2.5% C.I.	97.5% C.I.
Stimulus	.204	.452	.398	.517
Participant by Question Type	.019	.137	.100	.190
Participant by Brain Region	**.010**	**.099**	**.066**	**.143**
Participant	.037	.193	.118	.308
Residual	.175	.418	.404	.433

This is evidence for a person-dependent TMS effect that is highly heterogeneous (see Section 4.2). Any non-zero effect is important for a new intervention, and we place our result in context in Section 6. This result is particularly exciting, since while TMS has successfully been used to improve performance in other domains (see Section 2.3), **ours is the first study providing evidence that TMS can alter outcomes for programming tasks**. Our results argue for further exploration of using TMS (e.g., with protocol that strictly excites brain activity) to improve computing outcomes.

We also note that there is no evidence for a systematic effect from certain TMS conditions that improves or impairs programming ability relative to other TMS conditions. That is, while the effects of SMA and M1 stimulation on programming question response times are different ($p = 0.028$, see replication package for details), one is not overall better or worse at improving outcomes. This is expected from our protocol (which focused on demonstrating the possibility of any effect, not on positive-only effects).

We can interpret our result using a "difference-in-differences" approach from generalization theory [77]. Consider an arbitrary member of the population placed in two scenarios. In each scenario, they are presented with the same set of questions from our stimuli. We consider the subject's average response time in each scenario, with the set of questions large enough that residual variance is negligible. If the subject undergoes TMS stimulation to the same brain region in both scenarios, then the difference in average response times is zero (with probability 1). By contrast, if the brain regions stimulated differ, the "difference in differences" of log-transformed response times is 0.099, equivalent to a ratio of 1.10× between the two differences in raw response times.

5.4 RQ4: TMS and Self Perception

Following each TMS treatment, participants reported their subjective perception of the task difficulty, both in isolation and relative to the last session (if applicable), on a Likert scale. Overall, we find *no statistically significant evidence of differentiation in the subjective perception of participants* across all treatments and all question types ($p \geq 0.21$).

[2] As with RQ1, we log-transform to reduce right skew in raw times (skewness 2.21 → 0.17) and model residuals (2.26 → 0.11).

While we observed no differences in participants' subjective perception of task difficulty following treatments, we note that participants self-reports are generally not reliable [25], and TMS may still be influencing task difficulty without participant conscious perception. Conversely, the lack of perceived differences may remove a potential barrier to participant retention in future investigations of TMS-based treatments for software engineering (e.g., if TMS were to make tasks seem more difficult or frustrating, participants drop-out might be impacted, cf. [19, 33], etc.).

6 DISCUSSION

TMS has been shown to significantly improve or impair results in many other fields (see Section 2.3), and our results extend this to software engineering. Although the effect size noted in Section 5.3 shows that the treatment condition (i.e., the region of the brain that is stimulated by TMS) accounts for 2.2% of the variance we see in response times this is a more substantial report than it may first appear. Many papers on CS interventions do not include effect size in their results at all, or omit comparisons to a non-intervention baseline: in a 2018 review of 129 papers on pedagogical interventions in computer science, none included an effect size [94].

Some published results of interventions in computer science that do include such information may report similar variance in outcomes to our results. For example, one longitudinal study by Cooper *et al.* [19] considered a two-week, full-day workshop completed by high school students, of which 7.5 hours were devoted to spatial skills for a treatment group. They report that "the treatment group improved by an average score of 1.06 [out of 16 APCS questions], and that this was significant at the $p = 0.07$ level" [19, Sec. 4.2]. Although our approach uses a very different methodology and the results are not directly comparable, we note that that initial study provided the basis for subsequent studies of hundreds of students [8] and is associated with a 1-credit spatial skills course at a large university [92] where it improved retention in the major and grade outcomes for other classes. A small effect in an initial study may lead to a useful intervention later.

Additionally, since our main goal was to determine if neurostimulation could impact computing outcomes, we selected a protocol that may help *or* hinder ability. However, other TMS protocols exist that solely excite regions of the brain and/or otherwise observe predominantly positive results (e.g., [43, 46, 96], see Section 2.3). As a concrete example, an intermittent protocol (rather than the continuous protocol used for this experiment) involving applying theta burst TMS for 2s every 10s or a repetitive TMS (rTMS) treatment may result in heightened neural signal transmission [55]. Future work should investigate whether such heightened transmission translates to improved outcomes on computing tasks.

In addition to varying the protocol, future studies would benefit from varying the target brain area. While this paper investigated spatial skills, other studies implicate that brain regions associated with working memory (e.g., the dorsolateral prefrontal cortex) or language skills (e.g., Wernicke's or Broca's areas) may be correlated with other programming activities [33, 34]. Having demonstrated the applicability of TMS neurostimulation to computing outcomes, we encourage investigating positive impacts on other tasks.

7 THREATS TO VALIDITY

Since our presentation considers threats to validity throughout (e.g., from experimental design and execution to analysis), in this section we briefly summarize internal (e.g., did we apply TMS correctly?) and external (e.g., do our participants generalize to other populations?) threats, referencing earlier mitigation details.

Stimulation procedures. We adopt a well-established TMS protocol, supervised and approved by an outside TMS expert (Section 3.3.1). We individualized the treatment intensity (Section 3.3.2).

Brain region localization. We use fMRI brain scans for accurate brain region localization (Section 3.3.4).

Participant bias. We use an active stimulation control and did not convey expected effects (Section 3.3.3).

Tasks. We use stimuli validated in prior work and covering multiple distinct domains (Section 3.2). We acknowledge that the tasks considered may not generalize to other activities (e.g., pair programming) and defer such exploration to future research.

Population. Our participants (largely students) may not generalize to other populations (e.g., professional developers). We partially mitigate this by observing each subject longer, strengthening within-one-subject analyses (Section 3.3.1).

Training. We observe a statistically significant ($p < 0.01$) question type-dependent training/learning effect, which we account for in our data analysis (Section 4.2.3).

Subject variability. We use multi-level regression analysis, a well-established method to effectively account for between-subject heterogeneity (Section 4.2).

Researcher bias. We pre-registered hypotheses and methodology, conducted our preliminary analysis with anonymized labels, and corrected for multiple comparisons (Section 4.1).

Replication. Finally, our explicit replication of a previously-published [18] non-computing TMS result (the impact of SMA stimulation on mental rotation, Section 5.1) gives strong confidence in aspects of interval validity (i.e., applying TMS correctly).

8 RELATED WORK

In this section, we discuss other interventions impacting programming outcomes, contrasting them with TMS.

Neurostimulation represents a different, possibly orthogonal, mechanism for improving software developer abilities compared to standard approaches such as pedagogical structures (e.g., transfer training, tools, gamified or flipped classrooms), environmental factors and development methodologies at software jobs (e.g., work from home, Agile/Scrum), and the use of substances in software workplaces (e.g., Adderall, cannabis).

Pedagogy. Dozens of studies have investigated the benefits of the flipped classroom model (in which instruction/learning is completed externally and discussion is done during traditional lecture time to enforce concepts) in computer science pedagogy [41]. Similarly, gamified learning (in which elements of games, such as leaderboards or points, are used in class) has been studied to see how extrinsic rewards can motivate engagement of students [50].

There has been preliminary success with pedagogical interventions involving spatial reasoning and STEM outcomes [8, 20, 92]. Despite positive outcomes, pure spatial reasoning training in engineering or computer science educations has not been widely

adopted. We believe that a more rigorous understanding of why spatial reasoning cross-training improves behavioral outcomes, and its costs and benefits, would make it easier for institutions to adopt.

Work Structure. The structure of offices and work hierarchies has been an ongoing and evolving topic broadly in the field of computer science for many decades, especially with the express goal of improving company or individual programmer productivity [30]. For example, since the COVID-19 pandemic, working from home has become more relevant, and a survey of 3,634 software developers and managers from Microsoft found that 68% perceived they were just as, or more, productive working from home [40]. Similarly, pair programming, a key component associated with Agile development and some pedagogical methodologies, has been linked to higher satisfaction and learning outcomes, fewer bugs, and better communication between software engineers [6, 84, 101].

Medication. Many individuals program with the aid of psychoactive substances, citing enhanced abilities or the alleviation of symptoms. For example, recent surveys and interviews of 801 and 26 professional programmers (respectively) in software workplaces who use such substances found that many who use cannabis while programming do so for enjoyment, but also to enhance creativity or brainstorming, while many who use stimulant medications do so for perceived enhancements for focus and specific focus-intensive software tasks such as debugging [32, 66]. Although many substances may improve the abilities or health, administering them at a company level may have serious legal or health impacts (e.g., Adderall usage among people not diagnosed with ADHD has been linked to stress or the pressure to make tight deadlines [26]).

Intervention Summary. In contrast to such traditional interventions, TMS does not require the use of language, effort on the part of either a teacher or a student, or much time to use. If TMS is found to be effective in some capacity for computer science outcomes, it could be used as a non-pedagogical intervention in tandem with other instructional, structural, or medical interventions.

9 COSTS AND SUBJECTIVE EXPERIENCE

In this section, we outline the unique costs and considerations we encountered during our TMS study, emphasizing the differences from correlative studies that solely employ fMRI or fNIRS.

Recruiting. Unlike fMRI or fNIRS, TMS protocols may preclude subjects with a history of seizures or anxiety-related disorders, as well as those reporting a lack of sleep the prior night. However, TMS does not suffer from fNIRS data quality issues from hair types (cf. [48]). TMS causal studies require participants to attend multiple sessions (treatment and control) on different days. Subjectively, we found the multi-session constraint to be challenging for recruiting.

Time. For both TMS (e.g., applied quickly in advance, lasting up to an hour) and fMRI/fNIRS (e.g., typically measured over an hour-long session) the effective interaction duration per session is similar. Critically, however, TMS is not limited to 60-second stimuli (unlike fMRI or fNIRS, which use the BOLD signal and are thus limited by the hemodynamic response function [15]). We used short stimuli here for comparison to previous work, but future studies could use more complex programming tasks.

Cost. TMS and fNIRS offer cost advantages over fMRI in terms of both initial costs and operating costs. An institution with an fMRI lab often charges per scan (e.g., $500 per hour [39]); a TMS or fNIRS machine can typically be used for free if present.

Our base experiment cost was $2,200 ($125 per participant for reimbursement, $200 for electrodes); we elected to use high-quality fMRI localization (30 scan-minutes per participant, an additional $4,000). Future work may investigate the necessity of fMRI-quality localization for programming-related TMS treatments.

Training. Each research team member completed over 20 hours of training before being authorized to operate the TMS machine.

IRB. An Institutional Review Board or Ethics Board handles human study research at our institution. Depending on the review board's experience with neuroimaging or stimulation techniques, getting approval for a study with fMRI or TMS can require a substantial amount of time and effort. For reference, this TMS study involved a 24-page IRB application plus a 14-page consent form. Using fMRI to localize brain regions required an additional 4-page data protection and privacy plan (in the United States, brain scans are HIPAA-protected). From our first submission to approval, the IRB process took four months.

Lessons Learned. Subjectively, the most difficult aspects of the experiment were training and participant scheduling. Conducting thresholding sessions under time constraints and manually targeting the hand-held TMS magnetic coil required practice. Our multi-visit protocol amplified scheduling intersection challenges between researcher, TMS equipment and subject availability.

10 CONCLUSION

To the best of our knowledge, this paper is the first exploration of the *causal* relationship between software engineering and neural activity via Transcranial Magnetic Stimulation (TMS), a noninvasive technique well-established in the literature. Previous correlative findings have revealed intriguing connections between specific neural regions and programming tasks. These findings laid the foundation for enhanced understandings of expertise, pedagogy, and retraining. However, the absence of studies confirming the causal nature of these relationships has constrained their practical applications and interpretations in the real world.

We address causality by applying TMS treatment to 16 participants, directly targeting two indicative brain regions (M1 and SMA) known to exhibit correlative connections to programming tasks. We compare stimulation effects to participant performance on computing tasks, including data structure manipulation, mental rotation, and coding comprehension. We followed established, state-of-the-art TMS practices that were overseen by independent TMS experts. To mitigate bias, we used a special active control, pre-registered our hypothesis, conducted aspects of the experiment and analysis blinded, and correct for multiple comparisons.

We **replicate prior psychology results that TMS impacts mental rotation** (Section 5.1, $p \leq 0.02$) — supporting replication in science and giving confidence that we are applying TMS correctly. We **find no evidence of a simple causal relationship**: disrupting activity in M1 or SMA does not uniformly reduce outcomes on computing tasks (Section 5.2, $p \geq 0.22$) — results that do not agree with multiple previously-established correlations [48, 56, 70, 86] and suggest that *interpreting* cognition for CS is complex (cf. [53]).

Most critically, we **find that TMS has an effect on response time for data structure and code comprehension tasks**. TMS accounts for 2.2% of the variance in observed outcomes, a statistically-significant effect (Section 5.3). This **provides evidence that TMS (neurostimulation) can alter outcomes for programming tasks**. Neurostimulation is a distinct approach from traditional pedagogy (e.g., it does not require a shared language, or indeed any communication at all) and has produced positive outcomes in computing-related areas (e.g., creativity, mathematics, etc., Section 2.3). Now that TMS has been demonstrated to impact programming outcomes, we look forward to future work investigating, and making real, the potential benefits of neurostimulation for software engineering.

ACKNOWLEDGMENTS

We gratefully acknowledge the partial support of the NSF (2211749) and NIH (S10OD026738), and the Functional MRI Laboratory at the University of Michigan.

Additionally, we thank Dr. Taraz Lee, Dr. James Brissenden, and Quynh Nguyen for their help with TMS training, Dina Salhani with her scheduling and logistical assistance, and Dr. Kerby Shedden for his help with statistical analyses.

On a more minor note, our appreciation also goes out to Detroit Street Filling Station, a restaurant which sustained us for many days and fit all of our dietary restrictions.

REFERENCES

[1] A. Abadie. Causal inference. In K. Kempf-Leonard, editor, *Encyclopedia of Social Measurement*, pages 259–266. 2005.
[2] H. Akaike. Information theory and an extension of the maximum likelihood principle. *International Symposium on Information Theory*, pages 267–281, 1973.
[3] A. Alexander-Bloch, J. N. Giedd, and E. Bullmore. Imaging structural co-variance between human brain regions. *Nature Reviews Neuroscience*, 14(5):322–336, 2013.
[4] X. An and Y.-F. Yung. Item response theory: What it is and how you can use the IRT procedure to apply it. *SAS Institute Inc. SAS364-2014*, 10(4):1–14, 2014.
[5] D. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting linear mixed-effects models using lme4. *arXiv preprint arXiv:1406.5823*, 2014.
[6] A. Begel and N. Nagappan. Pair programming: What's in it for me? In *Empirical Software Engineering and Measurement*, page 120–128, 2008.
[7] C. M. Bennett, A. A. Baird, M. B. Miller, and G. L. Wolford. Neural correlates of interspecies perspective taking in the post-mortem atlantic salmon: An argument for multiple comparisons correction. *Neuroimage*, 47(Suppl 1), 2009.
[8] 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 *Computer Science Education*, pages 766–772, 2020.
[9] B. Boroojerdi, H. Foltys, T. Krings, U. Spetzger, A. Thron, and R. Töpper. Localization of the motor hand area using transcranial magnetic stimulation and functional magnetic resonance imaging. *Clinical neurophysiology*, 110(4):699–704, 1999.
[10] M. J. Burke, P. J. Fried, and A. Pascual-Leone. Chapter 5 - transcranial magnetic stimulation: Neurophysiological and clinical applications. In M. D'Esposito and J. H. Grafman, editors, *The Frontal Lobes*, volume 163 of *Handbook of Clinical Neurology*, pages 73–92, 2019.
[11] R. M. Byrne and P. N. Johnson-Laird. Spatial reasoning. *Journal of memory and language*, 28(5):564–575, 1989.
[12] L. Cárdenas-Morales, D. A. Nowak, T. Kammer, R. C. Wolf, and C. Schönfeldt-Lecuona. Mechanisms and applications of theta-burst rtms on the human motor cortex. *Brain topography*, 22:294–306, 2010.
[13] J. Castelhano, 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*, 13:623–637, 2019.
[14] J. E. Cavanaugh and A. A. Neath. The akaike information criterion: Background, derivation, properties, application, interpretation, and refinements. *Wiley Interdisciplinary Reviews: Computational Statistics*, 11(3):e1460, 2019.
[15] J. E. Chen and G. H. Glover. Functional magnetic resonance imaging methods. *Neuropsychology review*, 25:289–313, 2015.
[16] C.-N. Cheung, J. Y. Sung, and S. F. Lourenco. Does training mental rotation transfer to gains in mathematical competence? assessment of an at-home visuospatial intervention. *Psychological Research*, 84:2000–2017, 2020.
[17] A. Ciricugno, R. J. Slaby, M. Benedek, and Z. Cattaneo. *The Contribution of Non-invasive Brain Stimulation to the Study of the Neural Bases of Creativity and Aesthetic Experience*, pages 163–196. 2023.
[18] G. Cona, G. Marino, and C. Semenza. Tms of supplementary motor area (sma) facilitates mental rotation performance: evidence for sequence processing in sma. *Neuroimage*, 146:770–777, 2017.
[19] S. Cooper, K. Wang, M. Israni, and S. Sorby. Spatial skills training in introductory computing. In *International Computing Education Research*, pages 13–20, 2015.
[20] S. Cooper, K. Wang, M. Israni, and S. Sorby. Spatial skills training in introductory computing. In *International Computing Education Research*, pages 13–20, 2015.
[21] M. Cotelli, R. Manenti, S. F. Cappa, C. Geroldi, O. Zanetti, P. M. Rossini, and C. Miniussi. Effect of Transcranial Magnetic Stimulation on Action Naming in Patients With Alzheimer Disease. *Arch. Neurology*, 63(11):1602–1604, 11 2006.
[22] R. Cropanzano and T. A. Wright. When a "happy" worker is really a "productive" worker: A review and further refinement of the happy-productive worker thesis. *Consulting Psychology Journal: Practice and Research*, 53(3):182, 2001.
[23] P. Dassonville, S. M. Lewis, X.-H. Zhu, K. Ugurbil, S.-G. Kim, and J. Ashe. The effect of stimulus–response compatibility on cortical motor activation. *Neuroimage*, 13(1):1–14, 2001.
[24] A. C. Davison and D. V. Hinkley. *Bootstrap methods and their application*. Number 1. Cambridge university press, 1997.
[25] N. Dell, V. Vaidyanathan, I. Medhi, E. Cutrell, and W. Thies. "Yours is Better!": Participant response bias in HCI. In *Conference on Human Factors in Computing Systems*, page 1321–1330, 2012.
[26] A. DeSantis, S. M. Noar, and E. M. Webb. Speeding through the frat house: A qualitative exploration of nonmedical ADHD stimulant use in fraternities. *Journal of Drug Education*, 40(2):157–171, 2010.
[27] E. Diener and R. Biswas-Diener. The replication crisis in psychology, 2016.
[28] F. Duecker, T. A. de Graaf, C. Jacobs, and A. T. Sack. Time-and task-dependent non-neural effects of real and sham TMS. *PLoS One*, 8(9):e73813, 2013.
[29] J. Duraes, H. Madeira, J. Castelhano, C. Duarte, and M. C. Branco. Wap: Understanding the brain at software debugging. In *International Symposium on Software Reliability Engineering*, pages 87–92, 2016.
[30] H. Edison, X. Wang, and K. Conboy. Comparing methods for large-scale agile software development: A systematic literature review. *Transactions on Software Engineering*, 48(8):2709–2731, 2021.
[31] A. Eklund, T. E. Nichols, and H. Knutsson. Cluster failure: Why fMRI inferences for spatial extent have inflated false-positive rates. *Proceedings of the National Academy of Sciences*, 113(28):7900–7905, 2016.
[32] M. Endres, K. Boehnke, and W. Weimer. Hashing it out: a survey of programmers' cannabis usage, perception, and motivation. In *International Conference on Software Engineering*, pages 1107–1119, 2022.
[33] M. Endres, M. Fansher, P. Shah, and W. Weimer. To read or to rotate? comparing the effects of technical reading training and spatial skills training on novice programming ability. In *Foundations of Software Engineering*, pages 754–766, 2021.
[34] M. Endres, Z. Karas, X. Hu, I. Kovelman, and W. Weimer. Relating reading, visualization, and coding for new programmers: A neuroimaging study. In *International Conference on Software Engineering*, pages 600–612, 2021.
[35] 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*, 2018.
[36] J. J. Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press, 2016.
[37] E. Fast, C. L. Goues, S. Forrest, and W. Weimer. Designing better fitness functions for automated program repair. In *Genetic and Evolutionary Computation Conference*, pages 965–972, 2010.
[38] G. H. Fischer and I. W. Molenaar. Rasch models: Foundations, recent developments, and applications. 2012.
[39] B. Floyd, T. Santander, and W. Weimer. Decoding the representation of code in the brain: An fMRI study of code review and expertise. In *International Conference on Software Engineering*, pages 175–186, 2017.
[40] D. Ford, M.-A. Storey, T. Zimmermann, C. Bird, S. Jaffe, C. Maddila, J. L. Butler, B. Houck, and N. Nagappan. A tale of two cities: Software developers working from home during the OVID-19 pandemic. *Trans. Softw. Eng. Methodol.*, 31(2), 2021.
[41] M. N. Giannakos, J. Krogstie, and N. Chrisochoides. Reviewing the flipped classroom research: reflections for computer science education. In *Computer Science Education Research Conference*, pages 23–29, 2014.
[42] J. B. Gilbert, J. S. Kim, and L. W. Miratrix. Modeling item-level heterogeneous treatment effects with the explanatory item response model: Leveraging large-scale online assessments to pinpoint the impact of educational interventions. *J. Educational and Behavioral Statistics*, 2023.
[43] J. Gill, P. P. Shah-Basak, and R. Hamilton. It's the thought that counts: Examining the task-dependent effects of transcranial direct current stimulation on executive function. *Brain Stimulation*, 8(2):253–259, 2015.
[44] J. E. Gonzales and C. A. Cunninghham. The promise of pre registration in psychological research. *American Psychological Association*, 2015.

[45] K. Gustavson, T. von Soest, E. Karevold, and E. Røysamb. Attrition and generalizability in longitudinal studies: findings from a 15-year population-based study and a monte carlo simulation study. *BMC Public Health*, 12:1–11, 2012.

[46] E. Hertenstein, E. Waibel, L. Frase, D. Riemann, B. Feige, M. A. Nitsche, C. P. Kaller, and C. Nissen. Modulation of creativity by transcranial direct current stimulation. *Brain Stimulation*, 12(5):1213–1221, 2019.

[47] J. Hobot, M. Klincewicz, K. Sandberg, and M. Wierzchoń. Causal inferences in repetitive transcranial magnetic stimulation research: challenges and perspectives. *Frontiers in Human Neuroscience*, 14:586448, 2021.

[48] 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*, pages 396–407, 2019.

[49] Y.-Z. Huang, M. J. Edwards, E. Rounis, K. P. Bhatia, and J. C. Rothwell. Theta burst stimulation of the human motor cortex. *Neuron*, 45(2):201–206, 2005.

[50] M.-B. Ibanez, A. Di-Serio, and C. Delgado-Kloos. Gamification for engaging computer science students in learning activities: A case study. *Transactions on Learning Technologies*, 7(3):291–301, 2014.

[51] J. Jung, A. Bungert, R. Bowtell, and S. R. Jackson. Vertex stimulation as a control site for transcranial magnetic stimulation: a concurrent tms/fmri study. *Brain stimulation*, 9(1):58–64, 2016.

[52] S. Kang, J. Yoon, and S. Yoo. Large language models are few-shot testers: Exploring llm-based general bug reproduction. In *International Conference on Software Engineering*, pages 2312–2323, 2023.

[53] Z. Karas, A. Jahn, W. Weimer, and Y. Huang. Connecting the dots: rethinking the relationship between code and prose writing with functional connectivity. In *Foundations of Software Engineering*, pages 767–779, 2021.

[54] M. Klichowski and G. Kroliczak. Mental shopping calculations: A transcranial magnetic stimulation study. *Frontiers in Psychology*, 11, 2020.

[55] W. Klomjai, R. Katz, and A. Lackmy-Vallée. Basic principles of transcranial magnetic stimulation (TMS) and repetitive TMS (rTMS). *Annals of physical and rehabilitation medicine*, 58(4):208–213, 2015.

[56] 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.

[57] C. Lacadie, R. Fulbright, J. Arora, R. Constable, and X. Papademetris. Brodmann areas defined in mni space using a new tracing tool in bioimage suite. In *Meeting of the Organization for Human Brain Mapping*, volume 771, 2008.

[58] C. M. Lacadie, R. K. Fulbright, N. Rajeevan, R. T. Constable, and X. Papademetris. More accurate talairach coordinates for neuroimaging using non-linear registration. *Neuroimage*, 42(2):717–725, 2008.

[59] S. K. Lipson, S. Zhou, B. W. III, K. Beck, and D. Eisenberg. Major differences: Variations in undergraduate and graduate student mental health and treatment utilization across academic disciplines. *Journal of College Student Psychotherapy*, 30(1):23–41, 2016.

[60] P. L. Lockwood, G. D. Iannetti, and P. Haggard. Transcranial magnetic stimulation over human secondary somatosensory cortex disrupts perception of pain intensity. *Cortex*, 49(8):2201–2209, 2013.

[61] A. Mantovani, S. H. Lisanby, F. Pieraccini, M. Ulivelli, P. Castrogiovanni, and S. Rossi. Repetitive transcranial magnetic stimulation (rTMS) in the treatment of obsessive–compulsive disorder (OCD) and tourette's syndrome (TS). *International Journal of Neuropsychopharmacology*, 9(1):95–100, 2006.

[62] L. E. Margulieux. Spatial encoding strategy theory: The relationship between spatial skill and stem achievement. In *Conference on International Computing Education Research*, page 81–90, 2019.

[63] U. D. McCann, T. A. Kimbrell, C. M. Morgan, T. Anderson, M. Geraci, B. E. Benson, E. M. Wassermann, M. W. Willis, and R. M. Post. Repetitive Transcranial Magnetic Stimulation for Posttraumatic Stress Disorder. *Archives of General Psychiatry*, 55(3):276–279, 03 1998.

[64] S. Monteiro, G. M. Sullivan, and T. M. Chan. Generalizability theory made simple (r): an introductory primer to g-studies. *J. Graduate Medical Education*, 11(4):365–370, 2019.

[65] N. G. Muggleton, P. Postma, K. Moutsopoulou, I. Nimmo-Smith, A. Marcel, and V. Walsh. Tms over right posterior parietal cortex induces neglect in a scene-based frame of reference. *Neuropsychologia*, 44(7):1222–1229, 2006.

[66] K. Newman, M. Endres, W. Weimer, and B. Johnson. From organizations to individuals: Psychoactive substance use by professional programmers. In *International Conference on Software Engineering*, pages 665–677, 2023.

[67] I. Obeso, N. Robles, E. M. Marrón, and D. Redolar-Ripoll. Dissociating the role of the pre-sma in response inhibition and switching: a combined online and offline tms approach. *Frontiers in human neuroscience*, 7:150, 2013.

[68] J. P. O'Reardon, H. B. Solvason, P. G. Janicak, S. Sampson, K. E. Isenberg, Z. Nahas, W. M. McDonald, D. Avery, P. B. Fitzgerald, C. Loo, M. A. Demitrack, M. S. George, and H. A. Sackeim. Efficacy and safety of transcranial magnetic stimulation in the acute treatment of major depression: A multisite randomized controlled trial. *Biological Psychiatry*, 62(11):1208–1216, 2007.

[69] T. Paus. Inferring causality in brain images: a perturbation approach. *Philosophical Trans. Royal Society B: Biological Sciences*, 360(1457):1109–1114, 2005.

[70] N. Peitek, S. Apel, C. Parnin, A. Brechmann, and J. Siegmund. Program comprehension and code complexity metrics: An fMRI study. In *International Conference on Software Engineering*, pages 524–536, 2021.

[71] 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*, 66(3):260–264, 2008.

[72] D. Pizem, L. Novakova, M. Gajdos, and I. Rektorova. Is the vertex a good control stimulation site? theta burst stimulation in healthy controls. *Journal of Neural Transmission*, 129(3):319–329, 2022.

[73] A. G. Poydasheva, I. S. Bakulin, D. Y. Lagoda, A. A. Medyntsev, D. O. Sinitsyn, P. N. Kopnin, L. A. Legostaeva, N. A. Suponeva, and M. A. Piradov. Effects of online repetitive transcranial magnetic stimulation on the frequency of insights during anagram solving. In *Advances in Cognitive Research, Artificial Intelligence and Neuroinformatics*, pages 107–113, 2021.

[74] S. Pridmore, J. A. Fernandes Filho, Z. Nahas, C. Liberatos, and M. S. George. Motor threshold in transcranial magnetic stimulation: A comparison of a neurophysiological method and a visualization of movement method. *The Journal of ECT*, 14(1), 1998.

[75] Y. Qi, X. Mao, Y. Lei, Z. Dai, and C. Wang. The strength of random search on automated program repair. In *International Conference on Software Engineering*, page 254–265, 2014.

[76] N. Raman, M. Cao, Y. Tsvetkov, C. Kästner, and B. Vasilescu. Stress and burnout in open source: Toward finding, understanding, and mitigating unhealthy interactions. In *International Conference on Software Engineering*, page 57–60, 2020.

[77] D. B. Richardson, T. Ye, and E. J. Tchetgen Tchetgen. Generalized difference-in-differences. *Epidemiology*, 34(2):167–174, 2022.

[78] V. Romei, M. M. Murray, L. B. Merabet, and G. Thut. Occipital transcranial magnetic stimulation has opposing effects on visual and auditory stimulus detection: Implications for multisensory interactions. *Journal of Neuroscience*, 27(43):11465–11472, 2007.

[79] S. Rossi, M. Hallett, P. M. Rossini, and A. Pascual-Leone. Safety, ethical considerations, and application guidelines for the use of transcranial magnetic stimulation in clinical practice and research. *Clinical Neurophysiology*, 120(12):2008–2039, 2009.

[80] J. Rothwell. Techniques and mechanisms of action of transcranial stimulation of the human motor cortex. *J. Neuroscience Methods*, 74(2):113–122, 1997.

[81] F. Ruggiero, A. Lavazza, M. Vergari, A. Priori, and R. Ferrucci. Transcranial direct current stimulation of the left temporal lobe modulates insight. *Creativity Research Journal*, 30(2):143–151, 2018.

[82] A. T. Sack. Transcranial magnetic stimulation, causal structure–function mapping and networks of functional relevance. *Current opinion in neurobiology*, 16(5):593–599, 2006.

[83] A. Salguero, J. McAuley, B. Simon, and L. Porter. A longitudinal evaluation of a best practices CS1. In *Conference on International Computing Education Research*, pages 182–193, 2020.

[84] N. Salleh, E. Mendes, and J. Grundy. Empirical studies of pair programming for cs/se teaching in higher education: A systematic literature review. *Transactions on Software Engineering*, 37(4):509–525, 2011.

[85] E. Shihab, P. Thongtanunam, and B. Vasilescu. Mining software repositories. *IEEE*, 2023.

[86] 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 *International Conference on Software Engineering*, pages 378–389, 2014.

[87] 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*, pages 140–150, 2017.

[88] J. Silvanto and Z. Cattaneo. Common framework for "virtual lesion" and state-dependent tms: the facilitatory/suppressive range model of online tms effects on behavior. *Brain and cognition*, 119:32–38, 2017.

[89] J. Silvanto, Z. Cattaneo, L. Battelli, and A. Pascual-Leone. Baseline cortical excitability determines whether tms disrupts or facilitates behavior. *Journal of Neurophysiology*, 99(5):2725–2730, 2008.

[90] J. P. Simmons, L. D. Nelson, and U. Simonsohn. Pre-registration: Why and how. *J. Society for Consumer Psychology*, Dec. 2020.

[91] M. W. Sliwinska, S. Vitello, and J. T. Devlin. Transcranial magnetic stimulation for investigating causal brain-behavioral relationships and their time course. *JoVE*, (89):e51735, Jul 2014.

[92] S. Sorby, N. Veurink, and S. Streiner. Does spatial skills instruction improve stem outcomes? the answer is 'yes'. *Learning and Individual Differences*, 67:209–222, 2018.

[93] A. Suppa, Y.-Z. Huang, K. Funke, M. Ridding, B. Cheeran, V. Di Lazzaro, U. Ziemann, and J. Rothwell. Ten years of theta burst stimulation in humans: established knowledge, unknowns and prospects. *Brain stimulation*, 9(3):323–335, 2016.

[94] C. Szabo, N. Falkner, A. Knutas, and M. Dorodchi. Understanding the effects of lecturer intervention on computer science student behaviour. In *ITiCSE*

conference on working group reports, pages 105–124, 2018.

[95] I. Wagner. Gender and performance in computer science. *ACM Trans. Comput. Educ.*, 16(3), May 2016.

[96] H. Wang, M. Pan, C. Qiu, Z. Xue, X. Wu, and J. Tang. Research on digital cognitive behavior based on transcranial direct current stimulation. In *International Conference on Signal and Image Processing*, pages 1243–1247, 2021.

[97] E. M. Wassermann. Risk and safety of repetitive transcranial magnetic stimulation: report and suggested guidelines from the international workshop on the safety of repetitive transcranial magnetic stimulation, june 5–7, 1996. *Electroencephalography and Clinical Neurophysiology*, 108(1):1–16, 1998.

[98] W. Weimer, Z. P. Fry, and S. Forrest. Leveraging program equivalence for adaptive program repair: Models and first results. In *Automated Software Engineering*, pages 356–366, 2013.

[99] W. Weimer, T. Nguyen, C. L. Goues, and S. Forrest. Automatically finding patches using genetic programming. In *International Conference on Software Engineering*, pages 364–374, 2009.

[100] R. M. Willems, L. Labruna, M. D'Esposito, R. Ivry, and D. Casasanto. A functional role for the motor system in language understanding: Evidence from theta-burst transcranial magnetic stimulation. *Psychological Science*, 22(7):849–854, 2011.

[101] L. Williams and R. L. Upchurch. In support of student pair-programming. In *Technical Symposium on Computer Science Education*, page 327–331, 2001.

[102] N. Wong, V. Jackson, A. Van Der Hoek, I. Ahmed, S. M. Schueller, and M. Reddy. Mental wellbeing at work: Perspectives of software engineers. In *Human Factors in Computing Systems*, 2023.

[103] B. Xie, M. J. Davidson, M. Li, and A. J. Ko. An item response theory evaluation of a language-independent cs1 knowledge assessment. In *Technical Symposium on Computer Science Education*, pages 699–705, 2019.

[104] Y. Yamada. How to crack pre-registration: Toward transparent and open science. *Frontiers in Psychology*, 9(1831), 2018.

[105] S. Y. Yoon. *Psychometric properties of the Revised Purdue Spatial Visualization Tests: Visualization of Rotations (the Revised PSVT:R).* PhD thesis, 2011.