THE UNIVERSITY OF CHICAGO

MEMORY FOR ARITHMETIC FACTS LEARNED BY ROTE MEMORIZATION VERSUS VIA ALGORITHMIC COMPUTATION

A DISSERTATION SUBMITTED TO

THE FACULTY OF THE DIVISION OF THE SOCIAL SCIENCES

IN CANDIDACY FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF PSYCHOLOGY

BY

DEENA LYN BERNETT

CHICAGO, ILLINOIS

AUGUST 2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank a number of people who made this work possible. Thank you first and foremost to my advisor Susan Levine, who gave me tremendous latitude to pursue my own interests right from the beginning of my graduate career, and who was always supportive of any new project or idea that I wanted to tackle. I would also like to thank Daniel Casasanto, who mentored my trial research project in my first and second years of graduate school. I learned a number of invaluable skills under his guidance, and attending his lab meetings made me a better scientist. I would like to thank my committee members, Dave Gallo and Wilma Bainbridge, who provided insightful feedback and critical questions as I developed my plans for this dissertation. Finally, I would like to thank Steve Raudenbush, my outside reader, for taking the time to provide feedback on the final version of this dissertation and for generally serving as a positive role model and mentor during my time at the University of Chicago.

I would also like to thank several faculty members at other institutions who guided me to this point. Catherine Chase at Teachers College was my advisor for my master's degree and I continue to look to her as a model of a hard-working and dedicated researcher. Herb Ginsberg, also at Teachers College, inspired my interest in early math development and the classes that I took with him and his work informed much of my thinking on the subject. Kathy Hirsh-Pasek at Temple University first introduced me to the fascinating world psychological research when she mentored my high school senior project many years ago - an experience that directly led to my eventual decision to pursue a Ph.D. in psychology.

I would also like to thank all of the members of the Levine lab, too many to name, both past and present, who provided input that shaped my thinking on this work and who inspired me by sharing their own work over the past several years. Undergraduate students Anya Edelstein, Jacob Reiber, and Ella Marrero assisted with data collection on some earlier versions of these studies that,

although they did not make it into this dissertation, nonetheless were essential steps in refining my study design.

I would also like to thank the Institute for Educational Sciences, who provided generous financial support for my studies and research while at the University of Chicago, in the form of a Pre-Doctoral Fellowship, and the University of Chicago's Norman H. Anderson research fund, which provided funding for Study 1 of this dissertation.

Finally, I would like to thank my family for their tremendous support of me over the years and without whom this work would not have been possible. My husband, Ben, who has always been in my corner, has been a wonderful sounding board for thinking through ideas over the years, and has been incredibly supportive of my studies and research, assisting with family and other duties when I needed a bit of extra time to work. I would also like to thank my daughters, Olivia and Nora. In many ways this work was inspired by them and by a desire to help other students like them.

ABSTRACT

Does computing the answer to an arithmetic problem yourself help you to more fluently recall that answer? For example, if you are asked to find the answer to 8 x 6 by adding eight six times, do you remember that answer better than if you are simply *told* that 8 x 6 = 48 and directed to memorize it? Across four studies I pit learning artificial arithmetic facts (e.g., a # b = 10(a +b) + ab) via flash-card like rote memorization against learning those same facts by actually computing the answer. I found that, on a speeded, cued-recall test given immediately after practice, participants who practiced flashcard-like memorization consistently outperformed those who computed the answers themselves. This advantage persisted even after learning a second, interfering, set of facts. However, my third and fourth studies demonstrate that despite this initial advantage of memorization practice, computing the answer yourself may still result in better long term fluency, via a mechanism that I term "self-re-presentation". Essentially, the idea is that if you know how to compute a fact, then you can always recompute it when you cannot recall it, and, most importantly, that re-exposure to the correct answer will boost your subsequent ability to *recall* that answer. An additional layer of complexity is introduced by Study 2, which suggests that memorization alone *can* lead to robust long term memory provided that practice occurs with enough frequency that little forgetting occurs between practice sessions. This nuanced answer provides practical guidance for educators, clarifies why prior studies comparing self-computation to rote memorization have produced mixed results, and contributes to the human memory literature, describing an important learning mechanism (self-re-presentation) that has not figured prominently in prior discussions of memory.

Keywords: arithmetic fact, memory, computation, memorization

1. CHAPTER ONE: Introduction

1.1 Motivation

Elementary school students are expected to master single digit arithmetic combinations (e.g., 2 + 3 = 5, 7 x 8 = 56; National Council of Teachers of Mathematics, 2000; National Mathematics Advisory Panel [NMAP], 2008), and mastery of these combinations predicts success in secondary school mathematics (Geary, 2011; Price et al., 2013; Fuchs et al., 2006). Notably, it typically takes students upwards of four years to master these combinations, with some students failing to ever achieve complete mastery (Carpenter & Moser, 1984; Campbell & Graham, 1985). This difficulty is particularly striking considering how few unique arithmetic combinations a student needs to learn. If we ignore subtraction and division (which can be solved by reference to the corresponding addition and multiplication facts), exclude +0, +1, x0, and x1 facts (which can be solved by reference to a general rule), and do not count x2 facts (like 5 x 2) as distinct from doubles addition facts (like 5 + 5), there are only 36 unique addition combinations and 28 unique multiplication combinations that students must remember. As Campbell (1987) points out, it is truly astounding that it takes students years to master this small set of facts when in that same period they learn thousands of words, names, and other facts. This naturally raises the question of whether arithmetic fact mastery is doomed to be this painfully slow and difficult or whether instruction might be altered to promote faster and more complete mastery.

As we consider what type of instruction might best promote arithmetic facts mastery, one important instructional decision is whether students should practice by calculating the answers themselves or should engage in rote memorization in which the correct answers are given to them. As a concrete example, we might ask whether six year-olds should be given a page of addition problems and encouraged to find the answers by counting on their fingers, or whether they should be handed a stack of addition flashcards to memorize. Similarly, we might ask if it is better for eight year-olds to be encouraged to compute the answer to 6 x 4 by adding six four times or perhaps by

recalling 5 x 4 and then adding 4 more, or if they should instead spend their time chanting "six times four is twenty four." This debate over "meaningful memorization" or a "strategies approach" on the one hand, versus "rote memorization" on the other has persisted in the literature for nearly a century without a clear resolution (e.g., Brownell and Chazal, 1935; Hasselbring, Goin, & Bransford, 1988; Isaacs and Carroll, 1999; Walker et al., 2013; Baroody et al., 2016). Although the final answer to this question must necessarily consider a number of factors, both cognitive and affective (e.g., is one method more stressful than the other?), in this set of studies I attempt to shed light on one aspect of this debate, asking whether there is any mnemonic benefit to figuring out the answer yourself. That is, I ask: if a person is forced to calculate the answer to, say 12 x 13, themself, will they later remember that answer better than if they had simply been given the answer and directed to memorize it?

1.2 My Studies

I first preview my general methods as it will be helpful in understanding the relationship to prior studies. In four experiments, undergraduate students learned sets of six novel artificial arithmetic facts (e.g., 8 # 6 = 188) whose answers could be computed via a multi-step algorithm (a # b = 10(a + b) + ab) (see, for example, Reder and Ritter, 1992; Rickard, 1997 for prior studies using artificial arithmetic). Participants were assigned to one of two study conditions: self-computation or flashcard-like memorization (hereafter "computation" and "memorization"). The underlying algorithm was explained to the computation participants, and when each item appeared, they were required to generate the answer using the algorithm. For example, when they saw 8 # 3, they needed to add the two numbers (8 + 3 = 11), multiply that sum by 10 (11 x 10 = 110), multiply the two given numbers (8 x 3 = 24), and finally add the two pieces (110 + 24 = 134). By contrast, memorization participants were *not* given the underlying algorithm. Rather, to them, the facts appeared to be arbitrary associations to be memorized. For example, they were told that 8 # 3 = 134, but given no reason or rationale for that relationship. Memorization participants repeatedly

practiced the facts via a computer program that mimicked flashcard studying, while computation participants repeatedly practiced generating the answers. After 8-10 rounds of practice (depending on the study), participants in both groups completed a speeded test measuring their ability to recall the answer to each problem in under two seconds.

It is worth noting that I intentionally designed my computation and memorization conditions to correspond closely to methods that are actually used in education, as my goal was to provide data that would be useful for informing educational practice. Specifically, I wanted my memorization condition to be as close to studying with flashcards as possible, as flashcards (or flashcard-like computer programs) are frequently used by real students learning arithmetic facts. I wanted my computation condition to feel more like the open-ended practice that occurs when a student is handed a worksheet with a number of problems, and told to find their answer using a particular method. For example, in the popular Eureka Math curriculum (e.g., Kaufman et al., 2017), one lesson from the third grade curriculum directs students to solve a series of 6 x n problems by first solving 5 x n and then adding n, e.g., 6 x 3 = (5 x 3) + 3 (Great Minds, 2016). Because they were designed to closely imitate real-word study methods, my two conditions were not idealized to differ on only one dimension, but rather differ in multiple ways. Specifically, as detailed in the next section, these conditions differed in five ways that could impact memory.

1.3 Theoretically Important Differences Between Conditions

First, and perhaps most salient, the conditions differed in terms of generation. Memorization participants are initially *shown* the correct answer (i.e., the first time they "flipped" each flashcard). Computation participants *generate* the correct answer themselves, i.e., as the result of a computation. This difference between generating the correct answer and being shown the correct answer held not only for the first presentation of each fact, but was also the case whenever participants subsequently forgot an answer. For example, if a computation participant successfully answered 8 # 6 = 188, but then forgot this answer when the same fact appeared in a subsequent

round, they would have to recompute the answer, and their experience would again be one of self-generation. If a memorization participant forgot the answer, they would once again flip the flashcard, and would again be shown the correct answer. Overall then, both on the first presentation and on subsequent trials in which the answer was forgotten, computation participants always generated the correct answer themselves, inside their own minds, while memorization participants always read the correct answer in from the outside. This difference is particularly noteworthy because the "generation effect" tells us that information that learners generate themselves is better remembered than information that is simply given to them to read (Kane and Anderson, 1978; Slamecka and Graf, 1978). Indeed some prior studies comparing self-computation to other learning methods (e.g., rote memorization, calculator computation) have framed the primary difference as being one of generation (McNamara, 1995; Rittle-Johnson and Kmicikewycz, 2008; Pyke, LeFevre, and Isaacs, 2008; Pyke and LeFevre, 2011).

A second major difference was in how much the two learning methods encouraged retrieval attempts. A retrieval attempt occurs when a person actively tries to recall the answer, whether they are successful or not. For example, if someone asks you, "what was your first grade teacher's first name?" you will likely spend at least a second or two actively searching your memory for that piece of information. Regardless of whether or not you were successful, the act of actively *trying* to recall it constitutes a retrieval attempt. In the context of arithmetic facts, if I ask you, "what is 3 x 4?", you likely attempt retrieval, that is you *try* to recall the answer, and if you are like most adults, you likely succeed quite rapidly. By contrast if I ask you, "what is 26 x 45?" you likely do not attempt retrieval, since you are well aware that you do not have this answer memorized. Instead, you likely immediately begin the multi-step process of calculating the answer. In my studies, I expected that participants in my memorization condition would attempt retrieval more often since, as is typical of flashcard study, their instructions emphasized retrieval, telling them, "when each problem appears, try to recall and type the answer." That is, on each trial, a memorization participant likely thought,

"hmm… 8 # 3? What was that one?" for at least a moment before either succeeding or giving up and asking to see the answer. By contrast, reflecting typical self-computation practice, the computation instructions did not encourage either computation or retrieval. Instead participants were told that "at first you'll have to do the math to answer each problem. After some practice, you may just remember the answers. It's fine to do the math if you need to. It's fine to skip doing the math if you remember the answer." Therefore, on a given trial, a computation participant might not even have bothered to try to retrieve the answer from memory. Instead they might have simply immediately begun the calculation, feeling that this was a more than satisfactory way of completing the task. Indeed they may have immediately launched into the computation even if they *would have been* capable of recalling the answer if they had tried. As a result, overall, we should expect more retrieval attempts in the memorization condition than the computation condition.

Importantly, retrieval attempts, regardless of whether or not they are successful, are known to result in better memory, a phenomenon referred to as the "retrieval practice effect" or the "testing effect" (for reviews, see Rowland, 2014; Kornell and Vaughn, 2016). Therefore, we might expect that the increased number of retrieval attempts in the memorization condition would result in better memory for the studied arithmetic facts. In one particularly relevant study, Pyke and Lefevre (2011) had adults learn alphabet arithmetic facts (e.g., A + 1 = B, B + 2 = D). Participants were randomly assigned to self-generation, calculator-only, or a "retrieve-else-calculator" condition. Self-generation participants were directed to compute the answers themselves by counting through the alphabet, e.g., to find "M + 3" by saying "N, O, P". Calculator-only participants were given an onscreen calculator. They simply had to type "M" and "3" into the computer and the answer "P" would appear. Finally, retrieve-else-calculator participants were given two seconds to try to retrieve the answer to each problem, and, if they were unsuccessful, were then given the same onscreen calculator as the calculator-only group. The authors found that, on a subsequent recall test, participants in the "retrieve-else-calculator" condition outperformed participants in the calculator-

only condition (but not participants in the self-generation condition). On the one hand, the better performance on the retrieve-else-calculator condition as compared to the calculator-only condition is a clear indication that encouraging retrieval boosts memory for the studied information. On the other, the similar performance between the self-generation and retrieve-else-calculator conditions suggests that whatever benefits accrue from retrieval practice may not outweigh the benefits of self-generation.

A third key difference between the conditions is that, for the computation participants, the answers were logically connected to the problems via other known arithmetic facts. That is, for a memorization participant, 8 # 4 = 152 appeared to be just a random set of five unrelated digits to memorize. By contrast, for a computation participant, 8 # 4 was connected to 152 via two known facts: $8 + 4 = 12$ (which in our algorithm turns into 120), and $8 \times 4 = 32$ (which is then added to 120 to get 152). This experience of actually computing the answer oneself and seeing how it relates to the problem may help memorization participants' memory in multiple ways. First, it may help them to consciously narrow the range of possible answers (e.g., recognizing that 8 # 4 must necessarily be larger than 120, that it must have a 2 as its ones digit since $8 \times 4 = 32$, or that the algorithm necessarily implies that the answer to 8 # 4 must be greater than the answer to 8 # 3). When other candidate answers spring to mind, e.g., if they recall 188 (the answer to 8 # 6) and wonder if this might have been the answer to 8 # 4, they can consciously reject it using the above facts, e.g., here, because it does not have a two as the ones digit. Second, their experience executing the computation may prime the answer at a level below conscious reasoning. For example, seeing the digits 8 and 4 may automatically activate the answer 32 in memory due to a strong long-standing association between any two numbers and their product. That activation of the number 32 may in turn activate the final answer of "152" (either because of their superficial similarity or because they were frequently produced in close succession during execution of the algorithm).

Although the cognitive psychology literature does not often discuss this phenomenon in the context of arithmetic facts, it discusses an analogous phenomenon in the context of learning verbal material (e.g., learning word pairs like "COW - BALL"). These studies of "elaboration" generally support the conclusion that when participants must learn to associate a cue with a target response, providing them with additional information that explicitly relates the cue to the target boosts target recall. For example, relative to participants who simply read "the old man bought the paint," those who read "the old man bought the paint to color his cane," were subsequently more accurate in answering "which man bought the paint?" (Stein and Bransford, 1979; see also Rohwer, 1966). Cognitive psychologists further propose that this effect of elaboration operates via one or both of the mechanisms that I discussed above: "inferential redundancy", in which participants consciously use recalled bits of information to infer what the forgotten information must have been, and "network redundancy", in which having additional connections between the cue and target can directly facilitate retrieval of that target (Bradshaw and Anderson, 1982). It is not unreasonable to suppose that relating an arithmetic problem to its answer via a series of logical computations might boost memory in the same way that verbal elaborations boost memory for verbal material. That is, understanding how exactly 152 is generated from and, therefore, related to 8 # 4, may make it easier for the computation participant to later recall that answer. In fact, in the educational psychology literature, a computation based approach to learning arithmetic facts has often been called "meaningful memorization" (contrasted with "rote memorization") to highlight precisely this feature, i.e., the idea that if you *understand why* the answer is what it is, you will better remember *what* it is (e.g., Baroody, Bajwa, and Eiland, 2009).

A fourth important difference between my computation and memorization conditions is attention and, relatedly, intentionality. That is, in the memorization condition, participants' explicit goal was to remember the provided answer. This likely would cause them to focus on that answer, to perhaps repeat that answer multiple times (in an attempt to commit it to memory), and possibly to

deliberately develop mnemonic strategies intentionally designed to help them to recall the answer. For example, a participant might see "8 # 8 = 224" and say to themself, "8 # 8 is 224, 8 # 8 is 224, 8 # 8 is 224", or "8 # 8 =244, ok, I need to remember that 8 # 8 is 224 - well I see that the problem has two eights and the answer is 224, I can think that each eight is made of two fours, so 'two eights is two two fours'". By contrast, a computation participant may be so focused on executing the computation, thinking "8 plus 8 is 16, times ten, that's 160, and 8 times 8 is 64, so 160 plus 64 is 224," that they hardly pay any attention to the final answer at all, holding it in working memory just long enough to enter it into the computer. These differences in the amount of attention that is paid to the final answer, and in the amount of deliberate memorization effort directed at that answer, could certainly have differential effects on memory.

That said, how exactly attention and intentionally affect memory is unclear. On the one hand, explicitly directing participants to "try to remember" presented information usually has little to no effect on memory (Hyde and Jenkins, 1973; Craik and Tulving, 1975). On the other, it has often been proposed that the effects of self-generation or retrieval practice may be at least partly attributable to attention. That is, when participants have to actively generate or retrieve a piece of information, they necessarily have to focus more on what that piece of information is than when they passively read the same information (Kane and Anderson, 1978).

Finally, a fifth difference worth mentioning is in the amount of time that participants spent on each item. Computing an answer necessarily takes longer than recalling it. This means that computation participants were expected to take longer to complete each trial, translating into a longer learning phase overall, a longer delay between successive problems, and a longer delay between repeated presentations of the same problem. It is well known that the spacing of items during the learning phase can affect subsequent memory, with the most common finding being that greater spacing leads to slower initial acquisition but better long term memory (e.g., Bahrick et al., 1993).

Thus, overall, there were five noteworthy differences between my computation and memorization conditions: internal generation in the computation condition versus external presentation of the target responses in the memorization condition (generation effect), a higher propensity to attempt retrieval in the memorization condition (retrieval practice effect), more "meaningful" connections between the problem and answer in the computation condition (elaboration), greater attention to the answer itself in the memorization condition, and greater spacing in the computation condition (spacing effect). I want to emphasize that rather than viewing these multiple differences between conditions as confounds, I view them as integral features of two popular learning methods.

1.4 A Focus on Fluency

Importantly, the ultimate goal of basic arithmetic facts instruction is fluency (also called "automaticity") (Haring et al., 1978; Logan, 1988; NMAP, 2008). That is, we don't just want students to be able to figure out the answer, we want them to immediately know the answer. For example, a student who fluently knows their multiplication facts can immediately state that 9 x 4 is 36. Another student may be able to correctly answer the same question but only by computing 10 x 4 - 4, or perhaps by skip-counting, "9,18, 27, 36." While both students "know" the answer to 9 x 4, only the first has achieved fluency. This further complicates the predictions made above, as most studies of factors that affect "memory" (e.g. generation, elaboration, retrieval practice, etc.) do not differentiate between fluent and dysfluent knowledge.

To clarify how fluency factors into prior studies of memory, it is first important to note that the concept of fluency is not unique to arithmetic facts knowledge, but rather applies to a broad class of information. For example, spelling can also be fluent or dysfluent. Fluent spelling occurs when you immediately write the word correctly without thinking, while dysfluent spelling involves a deliberate process of sounding out the word, remembering a relevant spelling rule (e.g., "i before e"), or trying several alternative spellings before selecting the one that "feels" right. Other factual

knowledge can also be fluent or dysfluent. You may be able to immediately recite the names of the first ten American presidents or the first ten elements of the periodic table, or you may be able to laboriously reason through what these items must be, perhaps using some logical inference, or shuffling items to different positions before landing on your final answer. Similarly, you may immediately understand the meaning of an obscure word when you read it, or you may have to pause and think about its latin roots or about the mnemonic device that you used to learn that word in high school. In all of these examples, the key point is that a person can respond correctly without responding fluently. That is, accuracy does not equal fluency.

Importantly, prior studies of factors that affect "memory" tend to view memory as a unitary construct, i.e., assuming that either you remember something or you don't, completely ignoring the issue of fluency. For example, in one common study design, participants learn word pairs (e.g., "fish - frog") under study conditions that vary on some dimension of interest (e.g., perhaps reading intact pairs versus generating the second word, perhaps studying once a day versus once a week, etc.). Then later participants are given a cued recall test (e.g., "what word appeared with fish?") or a free recall test (e.g., "write down as many studied words as you can remember"). These tests are almost always completed without time pressure. As a result, these outcome measures cannot differentiate between a participant who sees the word "fish" and immediately and automatically knows that its paired associate is "frog" and a participant who sees the word "fish" and has to think for a few seconds, e.g., "oh, what was that one that went with fish? I remember thinking that it was also an animal that lived in water and that both started with 'f'... oh… frog". This difference is important. While both participants answer the question correctly, they do so in very different ways. The first response is fluent, while the second is not.

Why does fluency matter? For some types of knowledge, like recalling the names of the past U.S. presidents, it likely does not. That is, we may not care whether a student recalls the answer fluently or needs a few seconds to figure it out. However, for many types of knowledge, including

knowledge of word meanings, spelling, arithmetic facts, arithmetic procedures, and more, fluency absolutely matters (e.g., see reviews by Chard, Vaughn, & Tyler, 2002; Kuhn & Stahl, 2003 regarding the effect of reading fluency on reading comprehension; see also Vasilyeva, Laski, & Shen, 2015 regarding the effect of basic arithmetic facts fluency on complex arithmetic.). A student who can correctly translate Spanish vocabulary words into English on an unspeeded test, may not know them fluently enough to be able to comprehend spoken Spanish dialogue in real time. This has real implications for their Spanish abilities. Similarly, a student who *can* figure out the answer to 9 x 7 by adding 9 seven times or by subtracting 7 from 70 lacks fluency, and may not be able to successfully simplify 45/63, which requires them to immediately recognize both 45 and 63 as multiples of 9.

Above, I predicted that computation might lead to better memory than memorization as a result of generation or elaborative encoding. On the other hand, I argued that memorization might lead to better memory due to the retrieval practice effect. These predictions are further complicated by the fact that prior studies that have documented these effects have typically used unspeeded tests - counting *all* responses as correct, even if they are not fluent, but in the present set of studies I specifically measure participants' ability to respond rapidly (fluently). It should not be taken for granted that manipulations that improve general recall success will also improve fluency. For example, the retrieval practice effect may operate in one of the following two ways (among other possibilities). Retrieval practice may directly strengthen the association between the cue and target, perhaps as a result of activating the same neural pathways that will later be used for recall. If this is the case, we should expect retrieval practice to increase both accuracy and fluency. Specifically, it should make the answers not only correct, but correct and *fast*. Alternatively, retrieval practice may encourage the learner to develop mnemonics or elaborations that link the cue and target (e.g., Carpenter & DeLosh, 2006). The learner may then consciously rely on these mnemonics to retrieve the correct answer on the final test, resulting in responses that are accurate but dysfluent, i.e., correct but *slow*. (Similar arguments could be made for other memory effects discussed above, e.g.,

generation, elaboration, etc.). If the retrieval practice effect operates via the first mechanism, i.e., by directly strengthening the connection at a neural level, improving both accuracy and fluency, then we would expect it to still hold in the present set of studies, where the outcome measure is speeded recall. On the other hand, if the retrieval practice effect operates via the second mechanism, i.e., if participants must deliberately use conscious elaborations to infer the correct answer, improving accuracy but not fluency, then retrieval practice would not be expected to benefit memory on my speeded post test. Overall, then, because fluency is the ultimate goal of arithmetic facts learning, I use a speeded recall test, while prior studies of memory phenomena have indexed memory by un-speeded accuracy. This makes it even less clear how prior psychological theory maps onto the present set of studies and less clear which of my two conditions will result in better test performance**.**

1.5 My Outcome Measure

As mentioned above, the primary outcome measure in all of the studies reported in this dissertation was performance on a speeded post-test. Specifically, on each post-test trial, participants were shown a studied arithmetic fact (e.g., 8 # 3 = _____) and were given 2 seconds to type the answer. If they could not answer within two seconds, the computer automatically moved on to the next question and the item was scored as incorrect. This outcome measure was specifically selected because, as discussed in the preceding section, fluency (or "automaticity") is the ultimate goal of arithmetic fact learning. As such, I was specifically interested in how well participants could directly *recall* the answers after having practiced either self-computation or flashcard-like memorization (i.e., without recourse to reasoning, inference, or computation). I needed to ensure that my outcome measure captured successfully recalled answers only and did not allow participants to respond in some other way.

After considering several alternatives, I decided that the best way to achieve this was to impose a time limit so short that it would be nearly impossible to do anything other than directly

recall and type the answer. Two-seconds was selected because prior studies have found that executing a multi-step algorithm typically takes upwards of 3-5 seconds (depending on how many steps it has), while direct retrievals from memory typically take 1-2 seconds (Logan and Klapp, 1991; Rickard, 1997; Pyke and LeFevre, 2011). I also verified this assumption in my first study (see Chapter 2), by interspersing novel items with trained items on an additional speeded test and testing whether participants were able to correctly compute the answers to any of the novel items in the allotted two seconds.

I believe that this choice of outcome measure represents an important methodological contribution to the literature. Very few prior studies of arithmetic fact learning have used such a measure, and none, to my knowledge has used a two-second time limit. Theile (1938) did have a time limit for each item, but his was more a generous four-second time limit. This means that he may have mistakenly counted some rapid computations as correct retrievals, which should bias his results in favor of the computation participants. That is, suppose that, in actuality, a computation and a memorization participant can both directly recall exactly 70% of the studied facts. On their post test, they each respond correctly to the 70% of items that they recall. What then happens on the 30% of items that cannot be recalled? Presumably the memorization student can do nothing with these items. When you have memorized the answers, either you know them or you don't. By contrast, the computation student, may attempt to compute the answers that they cannot recall. Although they may not successfully compute the majority of these answers in under four seconds, if they successfully compute the answer on even 20% of un-recalled trials, that should boost their final score to 76% correct versus the memorization student's 70% correct, making it erroneously appear that computation students can recall more facts.

Even more troubling, a number of studies have instead chosen to give a time limit for the entire test rather than a time limit per item (e.g., complete as many questions as you can in three minutes; Woodward, 2006). This should yield a very noisy and imprecise measure of how many facts

a student can recall. First, it allows the computation students freedom to choose how they respond. Some computation students may feel more comfortable computing (e.g., they may like the security of "checking" their answers), and may choose to compute on every problem. For example, in Study 4 (see Chapter 5), when asked about any strategies that they used, one participant wrote, "one thing I did notice was that after repeatedly seeing the same numbers to the problems in the untimed test, I was able to answer quicker by skipping the [sic] some of the math, but I did not want to go off instict [sic] alone so I checked my answer in my head before submitting." This approach of always "checking" one's answers by doing the math should result in a high degree of accuracy on those problems, but a slow pace of completion and a low number of correctly answered items overall. These computation students would appear to know very few facts, when it is possible that if they had been pressed to recall the facts, they could have recalled a large number of them. A separate but equally problematic issue arises if students can recall very few facts and "finish early". For example, suppose that a particular computation student and memorization student both can directly recall only 30% of the facts on the test, and that it only takes them one minute out of the three allotted minutes to respond to those items. The memorization student can do nothing with the remaining items, and gains no additional points during the remaining two minutes. By contrast, the computation student may use the remaining time to compute the answers to several additional problems that they could not recall, inflating their final score. As these two examples illustrate, when a test with an overall time limit (rather than a per item time limit) is used, in some cases computation students may appear to know fewer facts than they actually do, while in others they may appear to know more. Depending on which of these occurs more frequently, this may introduce a systematic bias in favor of one condition, or, even in the absence of such a bias, it should increase the amount of noise in the data, making it more difficult to detect a true effect. It should also be noted that in some studies, the post test had no time limit at all, in which case the results should always be biased in favor of the computation participants, who will likely compute the answer to any problem that

they cannot recall (e.g., McNamara, 1995; Delazar et al., 2005; Rittle-Johnson and Kmicikewycz, 2008).

Finally another common, but I would argue flawed, outcome measure is to give participants unlimited time to respond to each item, but to then attempt to separate out recalled trials from computed trials after the fact. This separation is typically done using either (a) participant self reports (e.g., after each response a participant may be asked to report whether they computed or recalled the answer), or (b) reaction time (with fast responses, e.g., under 2 seconds, being deemed retrievals, and slower responses being assumed to be computations; Logan and Klapp, 1991; Cerella, Onyper, and Hoyer, 2006; Pyke and LeFevre, 2011; Baroody et al., 2015). The problem with this approach is that allowing the computation participants to compute and/or retrieve as they see fit, necessarily means that one ends up measuring when participants *choose* to retrieve, not when participants are *able to* retrieve. To see why this is a problem, suppose that in each participant's mind, there is some underlying "association strength" between a problem and its answer. It seems entirely plausible that the association strength required to produce the correct answer when you deliberately try to retrieve it might be lower than the association strength required to have the correct answer spring to mind without even trying. In other words, this is the difference between "I know it, it just takes me a moment to think of it" and "I know it so well that the answer pops into my head without any effort".

Importantly, in a computation versus memorization study, we would expect the memorization participants to actively attempt retrieval on every trial (since retrieval is the only way they can respond) and therefore to respond correctly at the lower threshold of retrieval strength (i.e., at the "I know it, but I have to think about it for a half second" level). By contrast, we might expect computation participants to default to computation, since this is what they are in the habit of doing, and to only switch to retrieval when the answer is immediately available without any mental search effort (i.e., at the higher association strength level of "the answer immediately 'pops' into my

mind"). If this is the case, even if the association strength between a problem and its answer are the same in both conditions, when we index item knowledge by self-reported retrievals, we are likely to erroneously conclude that memorization participants "knew" more items than computation participants. In other words, such a measure would be biased in favor of the memorization condition.

Additionally, this measure may be further biased because, as noted above, some computation participants may deliberately and consciously choose to "check their answers" by computing even when they are aware that they can retrieve them. This deliberate difference in strategy choice between different computation participants makes this measure not only biased but noisy, as some computation participants may switch to retrieval as soon as they feel they are able and others may prefer the comfort of "checking their answers" via computation.

My outcome measure addresses the above issues by forcing participants in both conditions to attempt retrieval (because the time limit is too short to do much else). Thus, I measure how many problems participants in both conditions *can* directly recall, without any additional noise from possibly computed answers in the computation condition, or any concerns that computation participants might not *choose* to recall even if they are able to. This aligns with my stated goal of testing the effects of these two learning methods on direct *recall memory* for the practiced answers, or, put another way, with measuring arithmetic fluency, and represents an important methodological step forward in the study of arithmetic facts memory.

I do recognize that this outcome measure has some degree of noise due to the fact that participants were required to type their answers (a feature which could be improved in the future by replacing typing with accurate voice recognition-software). That is, even when a fact was known, a participant may have made a mistake in typing it (i.e., a typo), and likely would not have had time to correct that typo in the allotted two seconds. To help address this issue, I had participants respond to each of the studied facts five times over the course of each speeded post-test, providing a more

stable measure of whether or not a participant actually knew a particular fact. Additionally, participants completed a simple typing measure (retyping three digit numbers that appeared on the screen) at the beginning of each study. This typing measure was used in two key ways. First, participants who did not meet a minimum level of typing proficiency were excluded from the study. This ensured that there were no participants whose ability to demonstrate their knowledge was severely limited by the requirement to type their answers. Second, even among included participants, typing ability (i.e., their score from this typing measure) was included as a control covariate in all subsequent analyses, helping to ensure that any condition differences that we observed were not due to differences in typing ability among the two groups.

1.6 Prior Empirical Work - Classroom Studies

Having discussed both my methods and how psychological theory might apply to the present set of studies at length, I now turn my attention to prior empirical work that has directly contrasted these two methods of learning. I will divide this discussion into two parts - first treating classroom-based studies which typically appear in the educational psychology literature, and then separately turning my attention to lab-based studies which more often appear in the cognitive psychology literature.

Several prior extended classroom-based studies have compared learning arithmetic facts via self-computation versus rote memorization. In these studies, elementary students typically practice arithmetic facts using a particular assigned method over the course of several days, weeks, or even months, and then complete a post-test to assess their fact knowledge. Importantly, to my knowledge, all of these studies have included additional differences beyond just the type of practice students engage in. For example, in addition to engaging in self-computation, one group might also receive instruction that explicitly highlights the relationships between different facts, e.g., noting how 6 x n relates to 3 x n, or the two groups might learn the facts in different orders, with one group learning a random selection of facts each week and another group learning facts in related groups (e.g., Theile,

1938; Swenson, 1949; Thornton, 1978; Carnine and Stein, 1981, Woodward, 2006; Fuchs et al., 2010; Baroody et al. 2016). These additional instructional differences make it difficult to disentangle the direct effects of memorization practice versus computation practice on memory. Additionally, as noted in the preceding section, all of these studies use outcome measures (e.g., un-speeded tests, participant self-reports, etc.) that may bias the results in favor of one condition or the other.

I should note that despite this lack of conclusive evidence, that the general consensus in the educational psychology literature is that computation-based practice is superior to rote memorization (e.g., Isaacs and Caroll, 1999; National Research Council, 2001; Baroody, Bajwa, Eiland, 2009). However, I am struck by how remarkably few published empirical studies actually demonstrate a direct advantage of computation practice on memory for arithmetic facts (even with additional condition differences and imperfect outcome measures as discussed above). Indeed, in reviewing nearly a century of literature on this subject, I was able to locate only four papers demonstrating an advantage of computation over memorization (Theile, 1938; Swenson, 1949; Carnine and Stein, 1981; Woodward, 2006). I was surprised to discover that a number of papers that are frequently cited in support of the superiority of self-computation are actually opinion pieces (Rathmell, 1978; Isaacs and Carroll, 1999; Baroody, Bajwa, and Eiland, 2009), observational studies (Brownell, 1944; Henry & Brown, 2008), or single-condition studies (Brownell and Chazal, 1935; Steinberg, 1985). Two other papers that are frequently cited in support of self-computation use a quasi-experimental design involving intact classrooms, but (mis)analyze their data as if each student is an independent observation, massively inflating the significance of their results (Thornton, 1978; Cook and Dossey, 1982). We should also expect that publication bias would largely exclude null results from the literature, but I did also find a few classroom studies that reported no difference between self-computation and rote memorization on memory for arithmetic facts (Baroody et al., 2014). Overall, then while the educational psychology literature seems to strongly favor computation practice over rote memorization, the truth is that very few studies actually find a measurable

advantage of computation on arithmetic facts memory, and even those that do, are not actually pure tests of this question.

I want to clarify, that I do not necessarily disagree with the general consensus that self-computation practice is superior. It is certainly possible that it is. Indeed, the elaboration literature from the cognitive psychology tradition makes a compelling argument as to why it should be. Rather, I am simply pointing out that despite this general consensus, the actual empirical evidence is quite sparse. This makes the present set of studies particularly important. I should also note that some studies cite other benefits of self-computation, e.g., benefiting transfer (i.e., the ability to solve similar novel problems) rather than directly benefiting memory for the trained facts (e.g., Baroody et al., 2015). Although they are beyond the scope of the present set of studies, these additional benefits most certainly should be considered when making practical instructional decisions.

1.7 Prior Empirical Work - Lab Studies

Although no extended classroom studies have directly compared self-computation to rote memorization (without additional differences), a number of shorter-term lab-like studies have focused solely on practice type. These studies typically involve interventions that take place over the course of a single session or a handful of sessions and have either adults or elementary students as participants, making them more directly comparable to the present set of studies. However, importantly, while my studies use flashcard-like memorization, many of these other studies have used a form of memorization in which participants simply read or copied each fact and its answer (McNamara, 1995; Pyke, LeFevre & Isaacs, 2008; Rittle-Johnson & Kmicikewycz, 2008), or where participants saw each fact and its answer first, held it in memory over a delay of 1-2 seconds, and then reported the answer (Delazer et al., 2005). Notably, the retrieval-practice literature suggests that these reading-based forms of memorization should be less effective than flashcard-like memorization (see Rowland, 2014 for a review). I was specifically interested in testing computation

practice against *flashcard-like* memorization because (a) both are commonly practiced in actual educational settings, and (b) there are strong theoretical reasons for believing that both might benefit memory (see discussion above). It doesn't provide much useful information about whether real students should engage in computation practice if we test it against a form of learning (i.e., simply reading the facts) that is both known to be relatively ineffective and rarely used.

Although most lab-based studies use a weaker reading-based form of memorization, I was able to find two studies that, like mine, have directly compared *flashcard-like* memorization to self-computation: Logan and Klapp (1991) and Cerella et al. (2006). Additionally, Pyke and Lefevre (2011) compared a self-computation condition and a retrieve-else-calculator condition in which participants were given two seconds to attempt retrieval and then used a calculator to find the answer if retrieval failed - resulting in an experience very similar to flashcard-like memorization. These three studies report mixed results with Logan and Klapp (1991) and Pyke and Lefevre (2011) finding no difference between conditions and Cerella et al. (2006) reporting an advantage for memorization. Notably, these results contrast sharply with the general consensus in the educational psychology literature that self-computation practice is better.

Although these studies are the most similar to mine of any I have seen, they also differ from mine in several potentially important ways. First, all three studies gave immediate corrective feedback to participants in both conditions, while I required computation participants to repeat the problem until it was correct. I believe that this use of immediate corrective feedback should minimize the condition differences, as computation participants might avoid actually computing, or might half-heartedly compute, and then, when the feedback appears, might simply memorize the provided answer for use on future trials. That is, they might adopt a strategy more focused on memorization, since they know that they will be shown the correct answer, regardless of whether they compute correctly or not. Additionally, none of these studies used a speeded post-test - instead using participants' reaction times and/or self-reports to identify those trials that were solved by retrieval

(as opposed to computation). As discussed above, there is reason to believe that this may bias the results in favor of the memorization condition. Overall, then, not only are the results of these lab-based studies mixed (and in contrast to the educational psychology results), their use of corrective feedback in both conditions and their choice of outcome measures may limit their validity.

1.8 Concluding Remarks

In sum, in both the educational psychology and the cognitive psychology literatures, there has been sustained interest in the relative benefits of self-computation versus rote memorization for learning arithmetic facts. However, to my knowledge, no existing study has conclusively determined which of these methods results in superior memory. This is due to the presence of additional condition differences in some studies, the use of potentially biased post-tests in others, and the use of weaker forms of memorization (i.e., reading based rather than flashcard-like) in still others. Even overlooking these issues, the results are decidedly mixed, with some studies reporting an advantage of self-computation, some reporting an advantage of memorization, and others reporting no difference.

Additionally, existing psychological theory does not make a clear prediction about which of these two methods should be superior. On the one hand, retrieval practice (i.e., as in flashcard-based studying) is known to be a potent driver of learning. On the other, the elaboration and generation effect literature both offer compelling reasons why participants in the computation condition may remember more of the facts that they study. Differences in the amount of attention and intentionality in the two conditions, and in the spacing in the two conditions may also affect memory. These predictions are further complicated by my specific focus on fluency here, i.e., an interest in not just whether you can *somehow* come up with the right answer if given unlimited time, but rather in whether you can *rapidly and directly retrieve* the answer from memory. This distinction between fluent and dysfluent knowledge has largely been ignored in prior studies of memory, but is

absolutely germane when considering arithmetic facts learning, where the ultimate goal is fluency. The present set of studies should, therefore, make an important contribution to the literature, attempting to finally provide a conclusive answer to the question, "does computing the answer to an arithmetic problem yourself improve your ability to recall that answer?"

2. CHAPTER TWO Study 1: Memorization versus Computation - Immediate and Delayed Effects

2.1 Overview

In Study 1, I asked two simple questions. First, I asked do you remember the answers to a small set of arithmetic facts better after (a) computing the answers to those facts yourself or (b) learning them via flashcard-like memorization? Second, I asked does the same pattern of results persist after learning a second set of facts? The motivation for the first of these questions was discussed extensively in chapter 1. The motivation for the second is briefly described below.

It has been well-documented that memory for individual items is impaired when additional items are learned, particularly if the items are similar, a phenomenon known as "interference". Furthermore, interference has been shown to operate in two directions. Memory for previously known information can be impaired by learning new information, representing "retroactive interference" (e.g., Osgood, 1949). For example, after learning a second foreign language (e.g., Spanish), you typically experience more difficulty recalling words from your first foreign language (e.g., French). Interestingly the effect also goes the other way. That is, memory for new information is impaired if other similar information has been previously learned (e.g., Keppel and Underwood, 1962). This "proactive interference" might occur, for example, when a veteran teacher has difficulty learning the names of their new students because they confuse them with similar-looking past students, or when you can easily recall the positions of the desired cards in your first game of "memory" (aka "concentration"), but have more difficulty in subsequent rounds because you can't remember, for example, if the apple was third from the left in *this* round or in a previous round.

Interference has been proposed to account for a large portion of the difficulty that students experience in learning arithmetic facts (see DeVisscher and Noël, 2016, for a review). One piece of evidence for this claim is the fact that wrong answers to arithmetic problems are typically the correct answers to similar problems. For example, Norem (1928) reported that 91% of 5400 observed

multiplication facts errors were correct answers to other multiplication facts, and 70% of those were answers to other problems that shared one operand, e.g., with a student perhaps giving the answer to 7 x 6 as the answer to 7 x 8 (see also Campbell and Graham, 1985). Additionally, also suggesting a role for interference, Campbell (1985) demonstrated that practicing one set of problems makes those problems' answers more likely to be incorrectly given as the answers to other related problems. More recently, De Visscher, Noël, and colleagues (De Visscher and Noël, 2013; De Visscher and Noël, 2014; De Visscher et al., 2015; Noël and De Visscher, 2018) have proposed that dyscalculic students, who struggle to learn arithmetic facts, do so because those individuals are particularly sensitive to interference in memory.

Given the large role that inference is hypothesized to play in memory for arithmetic facts, I wondered whether difference results might be obtained when a single small set of facts was learned, as opposed to when multiple sets of facts were learned, creating additional potential for interference. I hypothesized that I might observe both proactive interference on the second test and retroactive on the third test (retesting the first set of facts) and that participants in the memorization condition might experience more interference than participants in the computation condition, due to the fact that more interconnected material is known to be less subject to interference (e.g., Meyers et al., 1984; Radvansky and Zacks, 1991).

2.2. Method

This study was pre-registered on aspredicted.org (identifier #30013) on October 29th, 2019.

Participants

Self-described undergraduate students from the United States and the United Kingdom were recruited via Prolific and participated in exchange for $11 USD. An initial sample (n = 220) completed a screener (for $1 USD) that tested their typing skills and their ability to recall single digit multiplication facts. Participants who successfully passed the screener (n = 130) were invited to schedule an appointment to participate in the full experiment (at least 24 hours later and at most 4

weeks later). Of those who were invited, 89 participants completed the experiment. Three were excluded: one who had unreliable internet connections resulting in timing issues, and two who completed the experiment without experimenter oversight, leaving a final sample of 86 (53 female, 1 non-binary gender, mean age 21.95 years).

Design

Participants were randomly assigned to one of two learning conditions: computation (n = 43) or memorization (n = 43). Half of the participants in each learning condition first learned six facts that all had 7 as the first operand (7 # 2, 7 # 4, 7 # 5, 7 # 6, 7 # 7, 7 # 9) and then later learned six facts with 8 as the first operand (8 # 3, 8 # 4, 8 # 6, 8 # 7, 8 # 8, 8 # 9) (memorization n = 22, computation n = 21), while the other half learned the two sets of facts in the reverse order.

The memorization condition was designed to emulate flashcard-based study as closely as possible. Specifically, I wanted memorization participants to see the question, attempt to answer it, and then press a button to reveal the correct answer when they either (a) thought they had answered correctly or (b) decided that they didn't know the answer. The only difference from standard flash-card based practice was that (in both the flashcard and computation conditions) participants saw the correct answer for a fixed amount of time rather than being able to review the correct answer for as long as they liked. This was done to minimize additional noise due to different participants choosing to spend different amounts of time studying the answers.

For the computation condition, I considered three designs. The first, which would have been closest to worksheet-based computation practice, would have given participants *no feedback* on their answers. I decided against this approach because it clearly disadvantages the computation participants if their errors go uncorrected. Second, I considered showing computation participants the correct answer immediately after each incorrect response. I decided against this option for two reasons - one practical, and one theoretical. On a practical level, I worried that this might decrease the amount of effort that participants put into computing. Indeed, some participants might skip the

computation entirely, enter a random answer, wait for the correct answer to be shown, and then memorize that answer for the next round. This would greatly diminish the differences between our two conditions, essentially turning the computation condition into a memorization condition for participants who adopted this approach. Additionally, a theoretical reason that I decided against this option was that I was particularly interested in ensuring that the correct answers came from an external source in the memorization condition and that the correct answers were generated internally for the computation participants as the "generation effect" literature suggests that this may impact memory (Slamecka and Graf, 1978). If computation participants were shown the correct answers when incorrect, then it would no longer be the case that all of their answers were generated internally.

Ultimately, I settled on a third option: requiring computation participants to re-do items until correct. That is, after computation participants responded to a question, if they were correct they advanced to the next question, and if incorrect they repeated that same question. If they answered incorrectly a second time, they repeated the question yet again, and continued repeating it until it was answered correctly. This is clearly different than typical paper-based practice, but might reasonably be the feedback structure in a computer-based arithmetic game, where a student is simply instructed to "try again" each time they answer incorrectly. This design (a) incentivizes participants to actually execute the computation (because they can't continue until they get it right), (b) ensures that participants in both conditions see the correct answer exactly once per item, and (c) means that all of the computation participants' correct answers are generated internally. However, I acknowledge that this introduces two potential additional differences between the memorization and the computation conditions. First, it is possible that computation participants will see more wrong answers than memorization participants. That is, a computation participant might potentially answer a question incorrectly two or three times before finally seeing the correct answer. By contrast, a memorization participant can only respond incorrectly once before being shown the correct answer.

Second, computation participants will likely take longer to complete each trial. While this is partly an inherent difference between flashcard-like practice and computation practice, as it necessarily takes longer to compute an answer than to recall it, it is also partly a result of my requirement that computation participants retry a problem when incorrect. I address both of these potential confounds in my analyses.

Materials and Apparatus

Participants completed the experiment remotely from their own computer. The experiment was hosted on the Gorilla Experiment Builder (www.gorilla.sc; Anwyl-Irvine et al., 2018), and appeared in a web browser in fullscreen mode. An experimenter supervised the participant via Zoom during the main portion of the experiment (but not the screener). Participants kept their video and microphone turned on and shared their screen throughout the experiment. The experimenter muted herself and turned off her video so as not to distract the participants or to lead them with her facial expressions, but let the participants know that she would be watching them and that they could talk to her if they ran into any issues. Participants were instructed to sit in a quiet room without distractions and were told that they were not allowed to write anything down, use a calculator, or ask anyone to help them complete the tasks. Since not all participants were using the same keyboard, the instructions stated that, for fairness, participants were not to respond with their numeric keypad if they had one, but rather were to use the number keys "at the top" of their keyboard.

Procedure

Consent and Getting Ready. Participants first completed an onscreen consent form that covered both the screener and the full experiment. Participants then completed a checklist of general directions for getting ready such as ensuring that they were in a quiet room and closing other applications that would trigger alerts.

Typing and Multiplication Screener. Participants next completed a brief screening measure consisting of two tasks: a typing assessment and a single-digit multiplication test. They first saw instructions for the typing measure. All on-screen instructions in this study were paced by the computer, with one sentence appearing on the screen at a time. For each typing item, a three-digit number appeared on the screen (e.g., 341 = ___), and participants were tasked with re-typing the same number within two seconds (e.g., 341 = 341). The typing items were 30 randomly selected three-digit numbers (Appendix B). Participants who correctly answered at least 21 of the 30 typing items advanced to the multiplication test; those who did not were dismissed and paid for their time. As described in Chapter 1, the purpose of the typing measure was to (a) exclude participants whose ability to demonstrate their knowledge would be severely limited by their typing ability, and (b) measure typing ability so that it could be controlled for in subsequent analyses.

The multiplication test also began with on-screen, computer-paced instructions. On each multiplication item, a single-digit multiplication fact appeared in horizontal format (e.g., 9 x 8 = ____) and participants were given two seconds to type the product. The multiplication items consisted of all single-digit multiplication facts, presented in both commuted orders (e.g., both 8 x 9 and 9 x 8), except for facts that had 0, 1, 2, or 5 as one of the operands, which were excluded for the sake of time and because earlier data suggested that they were too easy and didn't discriminate between participants. Additionally, three warm-up items (1 x 4, 10 x 5, and 4 x 10) appeared at the start of the multiplication test (see full list of multiplication items in Appendix C). Participants who correctly responded to at least 16 of the 39 multiplication items were immediately notified that they had "passed" the screener and were transferred to a scheduling website where they could sign up for an appointment to complete the full experiment (at least 24 hours later and at most 4 weeks later). Like the typing measure, the multiplication test was included partly to be used as a control measure in subsequent analyses. Additionally, it was used to exclude participants with poor multiplication

proficiency (below 40% correct), as pilot data suggested that when these participants were assigned to the computation condition they were unable to complete the study within the allotted time.

The following details pertain to both the typing measure and the multiplication test. The backspace key was operational and could be used to correct typos if time permitted. Each item disappeared after two seconds, regardless of whether or not the participant had responded. No feedback was given. A blank screen appeared for 1.5 seconds between successive items. Items appeared in a fixed random order, i.e., in the same order for all participants across both conditions.

Core Experiment Instructions and Set Up. When participants returned for their scheduled appointment, they again began by reading and checking off a series of "getting ready" steps. They then joined a video call via Zoom with an experimenter who reviewed the general instructions for the experiment, reminding them, for example, that they were not allowed to write anything down or ask anyone else to help them.

First Training. The first training opened with onscreen instructions specific to the participant's assigned condition. The memorization instructions emphasized that the computer would teach the participant "artificial arithmetic" facts using a method that was "similar to flashcards." Specifically, the participant was directed to read the problem and to try to recall and type the correct answer (or to type "n" if they had no idea) and to then press enter to reveal the correct answer. Participants had unlimited time to respond. The instructions emphasized that the same problems would appear over and over, and that, after the correct answer was shown, the participant should try to "remember it for next time." The underlying algorithm was not shared with memorization participants.

The computation condition instructions gave participants the underlying algorithm, a # b = 10(a + b) + a x b, and walked them through solving a few example problems. These example problems did not reappear in any later training or test. Computation participants were told that when each problem appeared, they should compute the answer and then press enter to submit it.

Like the memorization participants, they were forewarned that the problems would repeat. Computation participants were specifically told that they were not *required* to use the algorithm each time - if they remembered the answer they could simply type it from memory.

The instructions also described the nature of the feedback that participants could expect. Participants in both conditions saw a green check if their answer was correct and a red X if it was incorrect. Following a correct response, participants in both conditions moved on to the next question. However, following an incorrect response, memorization participants were shown the correct answer (for 1 second), while computation participants were required to redo the problem until it was correct. In both conditions, instructions focused on describing the *training* trials, and participants were not forewarned that the training would be followed by a speeded test.

The on-screen instructions were immediately followed by eight blocks of practice, with each of the six problems appearing once per block. Blocks were not demarcated in anyway; to the participants, the training appeared to be an unbroken succession of forty-eight trials. The order of the six problems in each block was randomized with the constraint that where the end of one block met the beginning of the next, repetitions of the same problem were separated by at least three other problems. This fixed random order was pre-determined and was identical for all participants (across both conditions). In both conditions, problems appeared on the screen in horizontal format, one at a time, and participants had unlimited time to type their answer. Successive problems were separated by a blank screen for 1.5 seconds.

First Speed Test. In both conditions, the training was immediately followed by a "speed test" - my primary outcome measure of interest. This test was accurately described to participants as testing the same artificial arithmetic problems that they had been practicing, but with a format similar to the multiplication pre-test, i.e., with a two-second response deadline and no feedback. The speed test consisted of five blocks with each of the six problems appearing once per block, for a total of 30 trials. Like the training, the blocks in the speed test were not demarcated, problems in

each block were randomized with the constraint that repetitions of the same problem were separated by a minimum of three problems, and the order of the problems was identical across all participants. Like the typing assessment and multiplication pre-test, each item was displayed for two seconds, successive items were separated by a blank screen for 1.5 seconds, and no feedback was given.

Second Training. The speed test was followed by a second training that was identical to the first except that a different set of problems was practiced. Participants who learned the six 7 # n facts in the first training, now learned the six 8 # n facts, and vice versa.

Second and Third Speed Test. The second training was followed by a second speed test, identical to the first, except that it now tested the problems learned during the second training. Immediately after the second speed test, participants completed a third speed test that was 100% identical to the first, i.e., re-testing their ability to recall the problems from the first training.

Fourth Speed Test. Finally, all participants completed a fourth speed test that had the same trial format as the other speed tests but that tested the trained 7 # n and 8 # n problems as well as novel 9 # n problems. This test consisted of 12 different problems (4 each from 7 # n, 8 # n, and 9 #n), and had a total of three blocks. Each of the 12 problems was tested once per block in random order, again with the same constraint regarding repetitions. Again the blocks were not demarcated in any way and the order was the same across all participants.

Follow Up Questions. At the end of the experiment, participants completed a series of follow-up questions (Appendix E). These questions asked participants if they noticed any patterns (to rule out computation among participants in the memorization condition), to describe any strategies that they used, and whether there was anything else they wanted to share.

2.3. Analyses

The same general statistical model, with slight modifications, was used in all four studies reported in this dissertation. It is worth taking some time to describe this model in detail here as it is more complex than the models typically used in psychological studies.

In all of my experiments, the outcome measure of interest was performance on a speeded test (hereafter "speed test"). In a given speed test, each participant responded to six different questions, five times each. For example, in an experiment with 100 participants, this would result in 3000 individual observations in my data set (6 different questions x 5 repetitions of each x 100 participants). Whether or not each individual response (of the 3000) was correct is modeled as being a Bernoulli random variable with an underlying probability, p_i, that varies for each person by question cell (e.g., we model a different probability, p_i, for each of 600 cells, i.e., six different questions times 100 participants). That is, participant #37 is modeled as having a particular probability of answering 8 # 7 correctly, and this is different from their probability of answering 8 # 6 correctly and also different from participant #42's probability of answering 8 # 7 correctly. This person by problem cell probability, or rather the log odds of this probability, is modeled as being a function of:

1. The participant's assigned learning conditions, i.e., computation versus memorization ($LearnCondition_i$)
2. The participant's typing score (capturing typing ability, as well as basic processing and reaction speed) ($TypingScore_i$)
3. The participant's multiplication pre-test score (capturing additional individual differences not captured by typing score) ($PretestScore_i$)
4. The problem set to which the question belongs, i.e., 7 # n versus 8 # n ($FirstOperand_i$)
5. A random effect of the individual person (α_j) (i.e., above and beyond what we would predict based on the above, e.g., Participant #29 may be particularly likely to answer questions correctly)

6. A random effect of the individual problem being tested (α_k) (e.g., across participants "8 # 7 = _____" may be less likely to be answered correctly than "8 # 6 = ____").
7. A random interaction effect of participant and problem ($\alpha_{j,k}$) (e.g., Participant #15 may be particularly likely to answer "8 # 3 = ____" correctly, i.e., above and beyond what would be predicted by all of the above).

Note that the random effects (i.e., #5-7 above) are necessary to account for the complex correlation structure among observations in our data set. That is, it is incorrect to treat all 3000 observations as independent and identically distributed. Rather we must acknowledge that there are 30 observations per participant, which, in turn, are actually just five repeated observations of six different questions. Together the full model is specified as follows:

$$correct_i \sim Bernoulli(p_i)$$

$$logit(p_i) = \alpha_0 + \beta_{LC} \cdot LearnCondition_i + \beta_{TT} \cdot TypingScore_i + \beta_{PT} \cdot PretestScore_i + \beta_{FO} \cdot FirstOperand_i + \alpha_{j[i]} + \alpha_{k[i]} + \alpha_{j[i],k[i]}$$

$$\alpha_j \sim Normal\left(0, \sigma_j^2\right)$$

$$\alpha_k \sim Normal\left(0, \sigma_k^2\right)$$

$$\alpha_{j,k} \sim Normal\left(0, \sigma_{j,k}^2\right)$$

(Where observations, *i*, are nested within Participants, *j*, and Problems, *k*)

In parameterizing this model, learn condition (computation or memorization) and first operand (7 or 8) were entered as zero-one indicator variables, with zero representing the computation condition and the 7 # n problems. Pretest score and typing score were both entered as percent correct standardized (i.e., converted to z-scores among our sample). Slight variations on this model were used in some analyses (e.g., including an effect of test round when two or more different tests in the same experiment were compared, excluding the $FirstOperand_i$ term in studies where a single set of problems was learned, etc.), and will be described in more detail in the relevant chapters.

Although my model requires this degree of complexity in order to accurately capture the correlation structure in my data, models this complex are notoriously difficult to fit - a fact which may explain why psychologists have historically opted to fit simplified models (thereby sacrificing some degree of accuracy in their analyses). The only reliable way to fit a model with this degree of complexity is as a fully Bayesian model. Frequentist approximations, e.g., via maximum likelihood, typically cannot handle the degree of computational complexity needed to fit such a model and attempting to run them usually results in an error message (Muth, 2018).

Bayesian statistical analyses are still not standard practice in psychological science, and any reader who is not yet familiar with them is encouraged to read Appendix A, wherein I briefly describe how Bayesian statistics works in general and in particular how to interpret the results of my Bayesian models.

2.4. Results

Follow Up Questions

Two participants in the memorization condition reported figuring out the correct underlying algorithm. This was absolutely worth checking, and if we had had a large number of memorization participants who had figured out the underlying algorithm, it would have greatly minimized the differences between conditions and would have suggested that I needed to start over with a different study design. In my pre-registration, I stated that memorization participants who reported figuring out the underlying algorithm would be excluded from my analyses. Upon further reflection, however, I realized that this exclusion would violate sound principles of experimental design. The memorization participants who figured out the algorithm may differ from the other participants on some underlying characteristic(s) (e.g., intelligence) and, as the computation participants had no opportunity to demonstrate which of them would have figured out the algorithm on their own if given the opportunity, excluding participants on this basis would potentially introduce additional condition differences. For that reason, I have included these participants in my analyses. That said, as

there were only two such participants, I suspect that the decision to keep or exclude them likely has minimal impact either way.

First Speed Test Performance

On the first speed test, participants in the memorization condition correctly answered more items than participants in the computation condition, (54.2% correct versus 38.3% correct, Figure 2.1). To analyze performance on this test, the model described in section 2.3 was run. The results indicate a very high probability that memorization-based learning leads to better speeded test performance than computation-based learning (Posterior Parameter Estimate [PPE]: median = 0.98, median absolute deviation [MAD] = 0.42, $P(\beta_{LC} \leq 0) = 0.012$; Figure 2.2; note that all parameter estimates are on a logit scale and, as such, the reader should use caution in interpreting their magnitude, see Appendix A). This probability can be interpreted as meaning that there is only a 1.2% chance that computation-based learning actually results in speed test performance that is as good as or better than memorization-based learning.

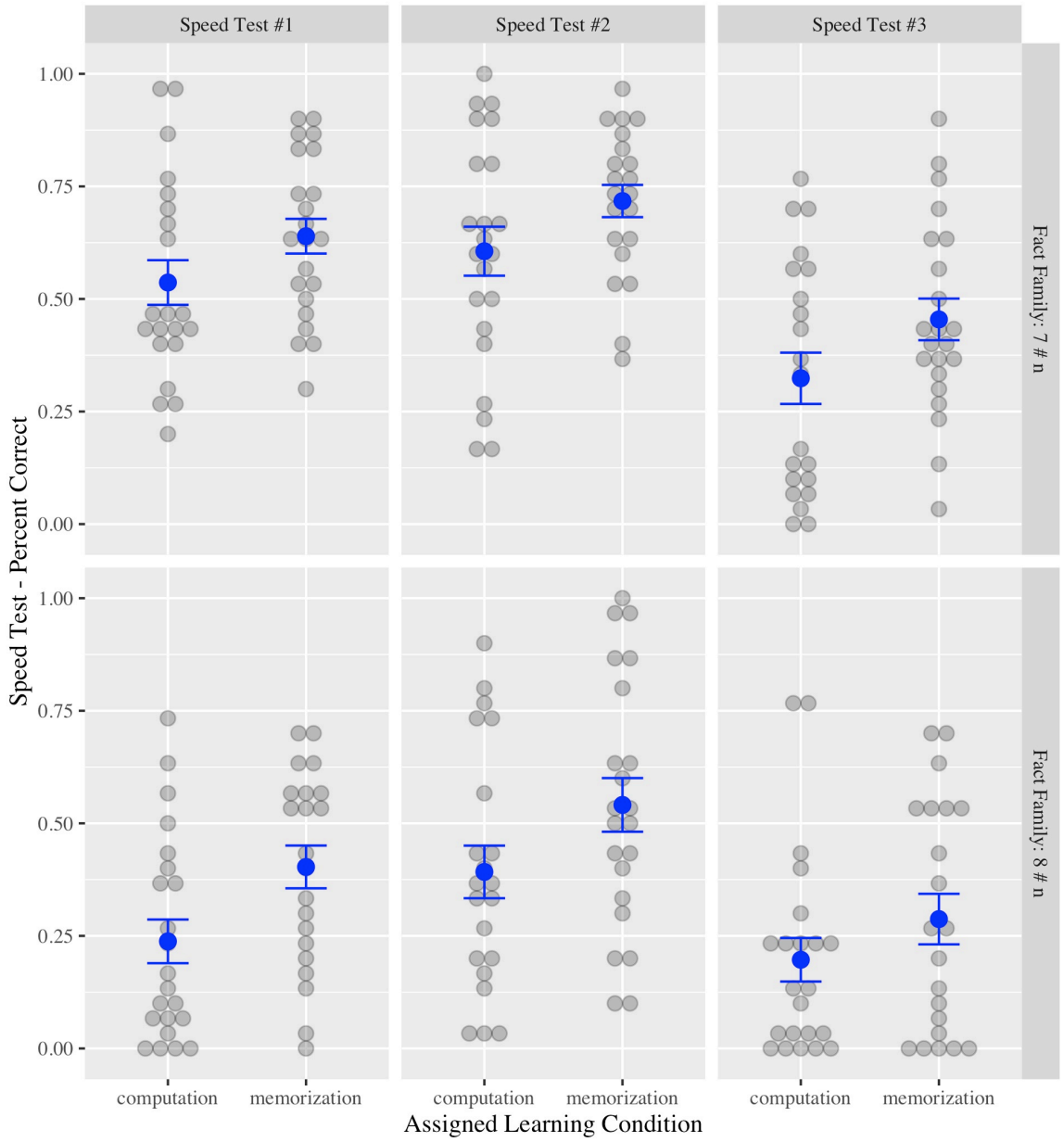

Figure 2.1. Performance on the three speed tests by assigned learning condition.
Each gray dot represents an individual participant's proportion correct on a speed test. Separated by participants who were tested on the 7 # n facts (top row) and participants who were tested on the 8 # n facts (bottom row). For transparency and ease of interpretation, blue dots and error bars represent simple averages and standard errors of the displayed data points (i.e., each participant's proportion correct). More precise estimates of the effect are provided by the Bayesian multi-level logistic regression model with control variables (see Figure 2.2), although the overall pattern of results is the same.

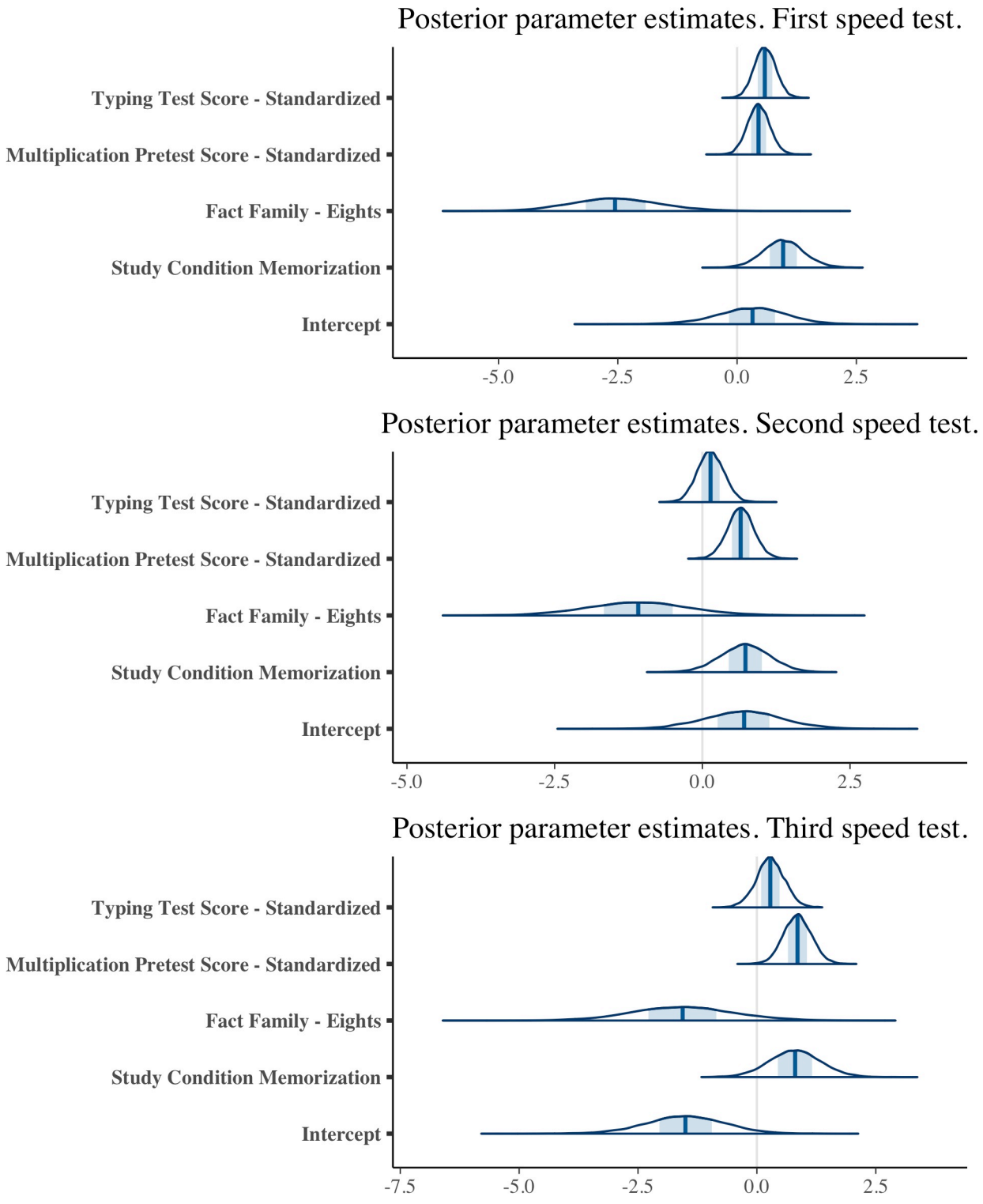

Figure 2.2. Posterior parameter estimates for models fit separately to the first, second, and third speed test data.
Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Study Condition Memorization".

Secondarily, I noted that participants could recall fewer of the 8 # n facts than the 7 # n facts (31.8% correct versus 58.9% correct; See Figure 2.1 and the Figure 2.2 "Fact Family Eights" parameter), and that participants who performed better on the typing and multiplication tests also performed better on the speed tests (see Figure 2.2 "Typing Test Score" and "Multiplication Test Score" parameters). Note that because typing and multiplication performance are included in the model, the effect of assigned learning condition is understood as being estimated while controlling for these factors.

Second Speed Test Performance

On the second speed test, memorization participants again outperformed computation participants (60.3% correct versus 50.2% correct; Figure 2.1). These data were analyzed using the same model used to analyze the first speed test data. Posterior parameter estimates again suggest that memorization results in better speed test performance than computation (Figure 2.2; PPE: median = 0.59, MAD = 0.22, $P(\beta_{LC} \leq 0) = 0.046$). We can interpret this probability as saying that, after learning a second set of facts, there is only a 4.6% chance that computation based learning actually results in speeded test performance that is as good as or better than memorization based learning.

First versus Second Speed Test Performance

To compare performance on the first and second speed tests, the following model was fit:

$$correct_i \sim Bernoulli(p_i)$$

$$logit(p_i) = \alpha_0 + \beta_{LC} \cdot LearnCondition_i + \beta_{TT} \cdot TypingScore_i + \beta_{PT} \cdot PretestScore_i + \beta_O \cdot Order_i + \beta_{TR} \cdot TestRound_i + \beta_{TR\times LC} \cdot TestRound_i \cdot LearnCondition_i + \beta_{TR\times O} \cdot TestRound_i \cdot Order_i + \alpha_{j[i]} + \alpha_{k[i]} + \alpha_{j[i],k[i]} + \beta_{j[i]} \cdot TestRound_i$$

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{\alpha_j}^2 & \rho_{\alpha_j\beta_j}\sigma_{\alpha_j}\sigma_{\beta_j} \\ \rho_{\alpha_j\beta_j}\sigma_{\alpha_j}\sigma_{\beta_j} & \sigma_{\beta_j}^2 \end{pmatrix} \right)$$

$$\alpha_k \sim Normal\left(0, \sigma_k^2\right)$$

$$\alpha_{j,k} \sim Normal\left(0, \sigma_{j,k}^2\right)$$

(Observations, *i*, are nested within Participants, *j*, and Problems, *k*)

I do not find a clear difference between the two learning conditions in the *change* in a participant's performance from test 1 to test 2 (PPE: median = -0.11, MAD = 0.36, $P(\beta_{TR\times LC} \geq 0) = 0.38$; Figure 2.3 and Figure 2.4's "Study Condition by Test Number Interaction" parameter). This largely reflects a *lack of certainty* about the size of the effect, rather than certainty that the effect is small or non-existent. Specifically the model predicts that a computation participant who learns 7 # n facts first (followed by 8 # n facts) will decline 1.5 percentage points on average from test 1 to test 2, whereas a memorization participant will decline 3.2 percentage points, for a difference between conditions of 1.7 percentage points on average. Importantly, estimates of the effect vary greatly (sd = 7.7 percentage points), indicating that anything in the range of -14 to 17 percentage points difference in the amount of change between conditions is reasonable. Estimates are even wider for a hypothetical participant who learns 8 # n facts first (mean difference between conditions in test 1 to test 2 change = +1.1 percentage points, SD = 11.2 percentage points).

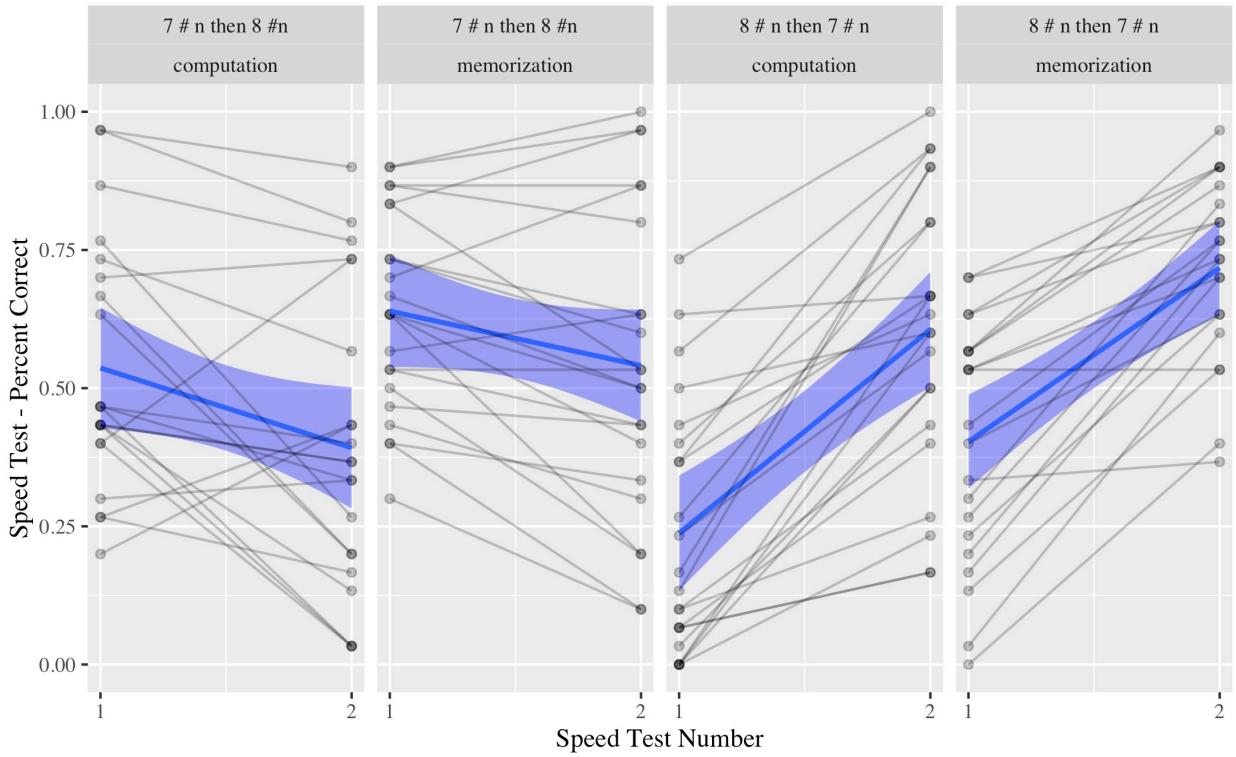

Figure 2.3. Change in performance from the first to the second speed test.
Data are separated by participants who learned the 7 # n facts first (left two panels) and participants who learned the 8 # n facts first (right two panels). Lines connect the two test scores belonging to the same participant. For transparency and ease of interpretation, blue lines and shaded error areas represent simple linear regressions based on the displayed data points. More precise estimates of the effect are provided by the Bayesian multi-level logistic regression model with control variables (see Figure 2.4), although the overall pattern of results is the same.

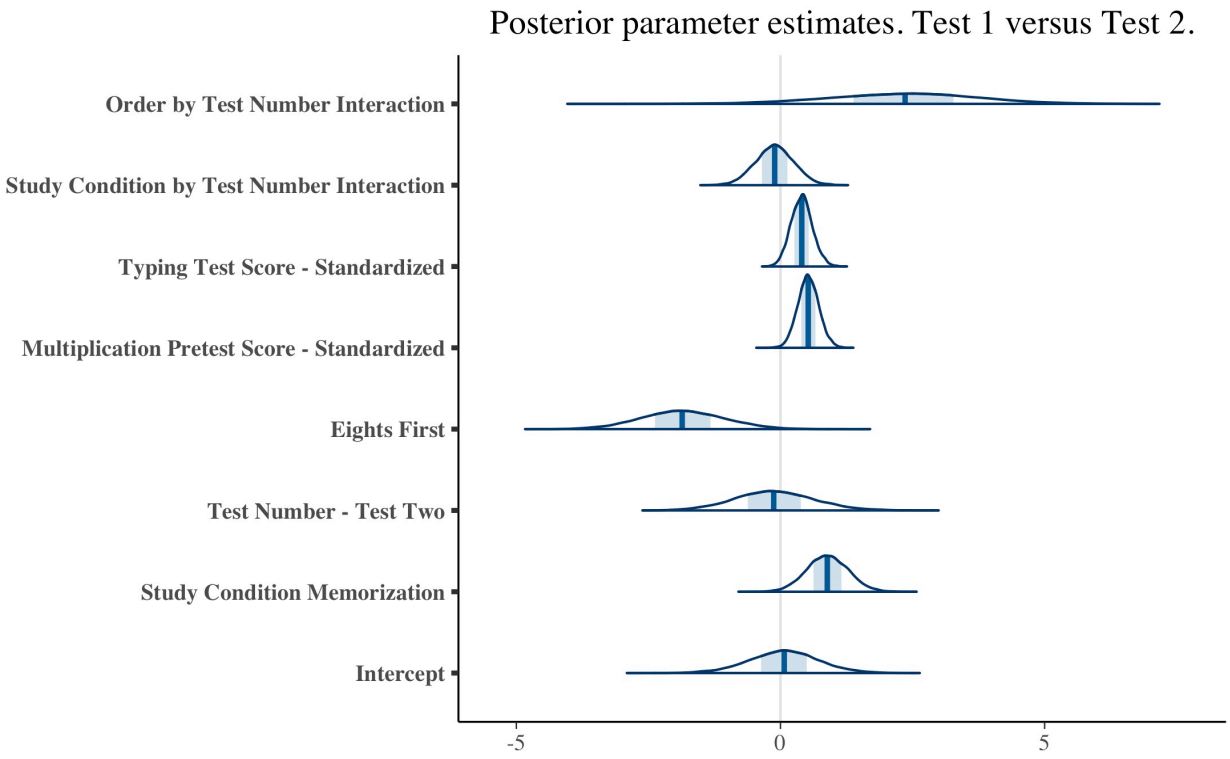

Figure 2.4. Posterior parameter estimates for the model comparing the first and second speed test data.
Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Study Condition by Test Number Interaction".

Secondarily, I note that participants perform better on the second speed test than on the first (56.4% correct versus 45.4% correct, PPE: median = 0.99, MAD = 0.18, $P(\beta_{TR} \leq 0) < 0.0001$; posterior not shown). Note that this effect was estimated using a modified version of the model above that did not include the $\beta_{TR\times LC} \cdot TestRound_i \cdot LearnCondition_i$ or the $\beta_{TR\times O} \cdot TestRound_i \cdot Order_i$ term, allowing me to recover the main effect of test round directly from the model output.

Third Speed Test Performance

The third speed test retested the first set of facts. Data from this test were analyzed using the same models as the first and second speed tests. Memorization participants once again outperformed computation participants (37.3% correct versus 25.9% correct; PPE: median = 0.80,

MAD = 0.54, $P(\beta_{LC} \leq 0) = 0.06$, Figures 2.1, 2.2), though, as can be see in the posterior distribution, we are less certain here that memorization is better than we were for the previous two speed tests, i.e., there is a 6% chance that computation is actually as good as or better than memorization on the third speed test.

First versus Third Speed Test Performance

The first and third speed tests were compared using the following model, which differed from the model used to compare the first and second speed tests because tests 1 and 3 tested the same set of facts, while tests 1 and 2 tested different sets of facts, e.g., if a participant was tested on 8 # n facts on test 1, they were tested on 7 # n facts on test 2, and then again tested on 8 # n facts on test 3.

$$logit(p_i) = \alpha_0 + \beta_{LC} \cdot LearnCondition_i + \beta_{TT} \cdot TypingScore_i + \beta_{PT} \cdot PretestScore_i + \beta_O \cdot Order_i + \beta_{TR} \cdot TestRound_i + \beta_{TR\times LC} \cdot TestRound_i \cdot LearnCondition_i + \beta_{TR\times O} \cdot TestRound_i \cdot Order_i + \alpha_{j[i]} + \alpha_{k[i]} + \alpha_{j[i],k[i]} + \beta_{j[i]} \cdot TestRound_i + \beta_{k[i]} \cdot TestRound_i + \beta_{j[i],k[i]} \cdot TestRound_i$$

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_j} & \rho_{\alpha_j\beta_j}\sigma_{\alpha_j}\sigma_{\beta_j} \\ \rho_{\alpha_j\beta_j}\sigma_{\alpha_j}\sigma_{\beta_j} & \sigma^2_{\beta_j} \end{pmatrix} \right)$$

$$\begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_k} & \rho_{\alpha_k\beta_k}\sigma_{\alpha_k}\sigma_{\beta_k} \\ \rho_{\alpha_k\beta_k}\sigma_{\alpha_k}\sigma_{\beta_k} & \sigma^2_{\beta_k} \end{pmatrix} \right)$$

$$\begin{pmatrix} \alpha_{j,k} \\ \beta_{j,k} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_{j,k}} & \rho_{\alpha_{j,k}\beta_{j,k}}\sigma_{\alpha_{j,k}}\sigma_{\beta_{j,k}} \\ \rho_{\alpha_{j,k}\beta_{j,k}}\sigma_{\alpha_{j,k}}\sigma_{\beta_{j,k}} & \sigma^2_{\beta_{j,k}} \end{pmatrix} \right)$$

(Observations, *i*, are nested within Participants, *j*, and Problems, *k*)

I found no evidence that the change in a participant's performance from the first to the third speed test differed by condition (median: 0.00, MAD: 0.36, $P(\beta_{TR\times LC} \leq 0) = 0.50$, Figures 2.5 and 2.6). Secondarily, I note that overall, participants performed worse on the third speed test than on the first (31.6% correct versus 45.4% correct; PPE: median = -1.40, MAD = 0.35, $P(\beta_{TR} \geq 0) = 0.0008$;

posterior distributions not shown). As noted for the comparison between tests 1 and 2, this main effect of test was recovered from a model without the $\beta_{TR\times LC} \cdot TestRound_i \cdot LearnCondition_i$ or the $\beta_{TR\times O} \cdot TestRound_i \cdot Order_i$ term.

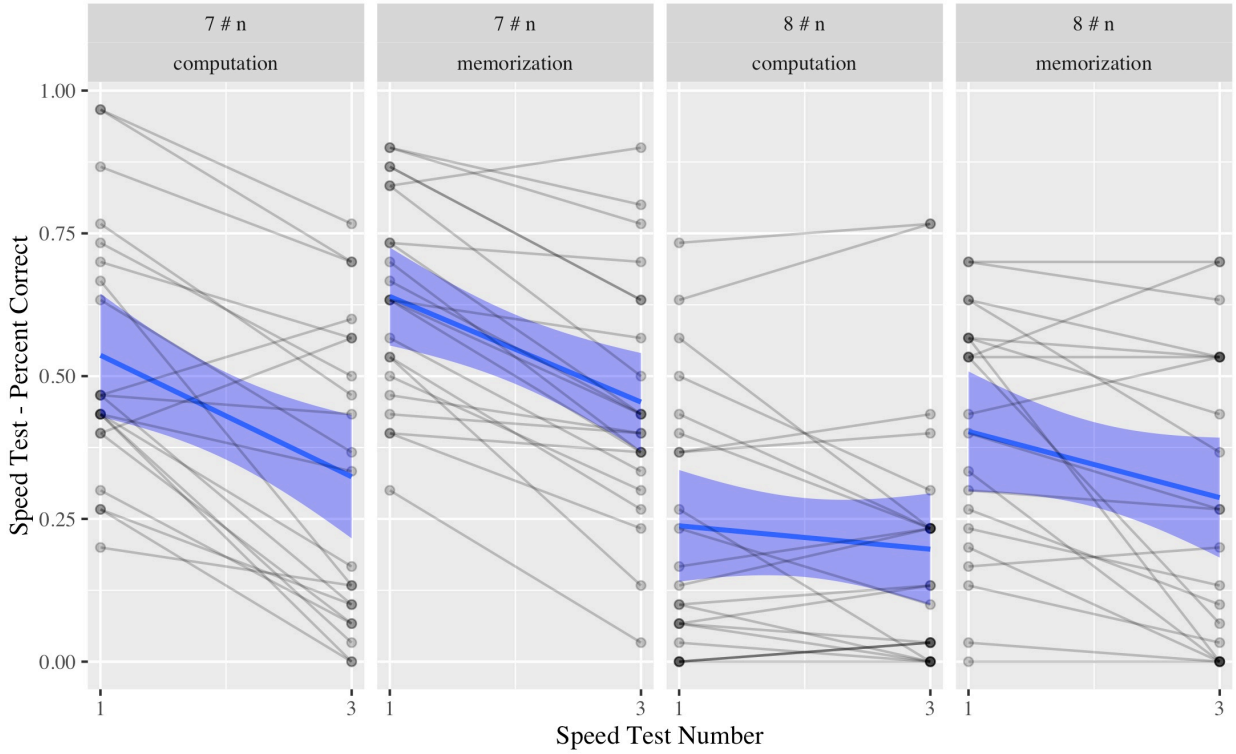

Figure 2.5. Change in performance from the first to the third speed test.
Separated by participants who learned the 7 # n facts (left two panels) and participants who learned the 8 # n facts (right two panels). Lines connect the two test scores belonging to the same participant. For transparency and ease of interpretation, blue lines and shaded error areas represent are simple linear regressions based on the displayed data points. More precise estimates of the effect are provided by the Bayesian multi-level logistic regression model with control variables (see Figure 2.6), although the overall pattern of results is the same.

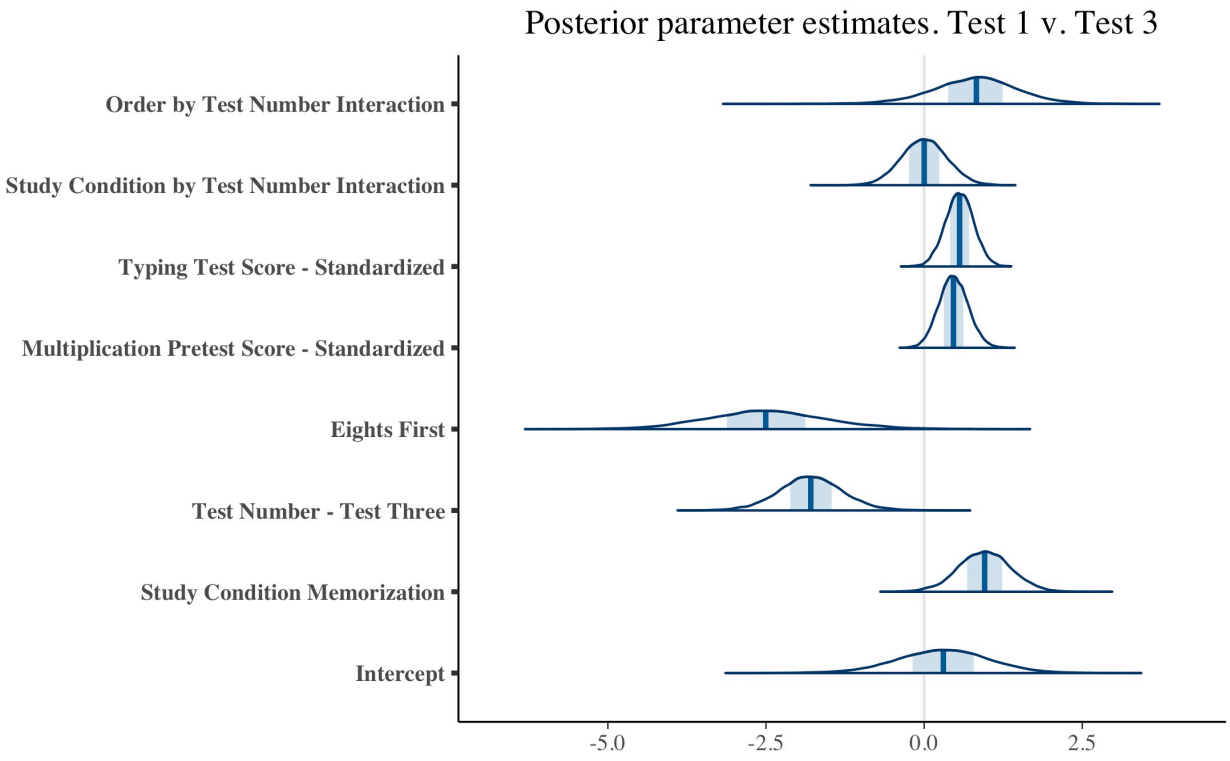

Figure 2.6. Posterior parameter estimates for the model comparing the first and second speed test data.
Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Study Condition by Test Number Interaction".

Again, my Bayesian analyses allow me to quantify the range of possible sizes of this effect. Specifically the model predicts that a computation participant who learns 7 # n facts first will decline 1.88 percentage points on average from test 1 to test 3, whereas a memorization participant will decline 1.97 percentage points, for a difference between conditions of 0.9 percentage points on average. Not only is this estimated difference quite small, but also estimates of the effect vary greatly (sd = 11.9 percentage points), indicating that anything in the range of approximately -23 to 25 percentage points difference between conditions is reasonable. Estimates are similarly quite wide for a hypothetical participant who learns 8 # n facts first (mean difference in test 1 to test 3 change between conditions = 2.0 percentage points, SD = 9.9 percentage points). Again, the take home

message here is not that we are certain that there is *no* difference by condition in the change from Test 1 to Test 3, but rather that we have very little certainty about whether or not this difference exists and, if it does, how large it is.

Fourth Speed Test Performance

The fourth speed test was included as a control measure, i.e., to check that on all of my speed tests, participants were in fact *recalling,* not computing, the answers in the two-second response window. On this test, participants responded to novel artificial arithmetic problems (9's facts) interspersed with the trained problems. I was specifically interested in whether participants would answer any of the novel problems correctly, indicating an ability to compute answers in under two seconds.

Only 2 of the 43 computation participants answered any of the novel problems correctly (as opposed to 36 who answered one or more trained problems correctly), and one of these two participants answered only one novel item correctly (out of 12) while the other correctly answered just two items. As expected, none of the memorization participants answered any novel item correctly. Overall, these results suggest that while it *is possible* for computation participants to answer the speed test items by computation, it is quite rare (occurring on just 3 out of 516 novel item trials). Although admittedly, computing the answer to a familiar problem (i.e., one whose answer you have computed before) may be faster than computing the answer to a novel problem. It is also worth noting that this source of bias serves to artificially inflate the performance of the computation group, suggesting that, if anything, the true difference in recall ability between computation and memorization participants is even larger than the difference I report.

Finally, I note that in my pre-registration, I specified that participants who answered the novel items correctly would be excluded from all analyses. Upon further reflection, for the same reasons given above as to why participants who figured out the underlying algorithm were not excluded, I did not exclude these participants (see "Follow Up Questions" section above).

Speed Test Performance By Block

I next examined speed test performance by block for the first speed test. We might wonder whether computation participants are simply caught off guard by the speed test format. Perhaps, being used to computing, they don't realize that the two seconds are really too short to compute and that they should only be trying to retrieve. Or perhaps they *can* retrieve the answers, but aren't very practiced at doing so (while the memorization participants have been practicing retrieval during their training). If these or other similar explanations are correct, we might expect to see computation participants performing particularly poorly on the first block (or even first two blocks) of the speed test, but greatly improving thereafter, reflecting the fact that they knew the answers, but just didn't initially understand the task demands. It is important to rule out the possibility that such an explanation accounts for the condition differences that I observe. To investigate this possibility, the following model was run using the data from the first speed test:

$$logit(p_i) = \alpha_0 + \beta_{LC} \cdot LearnCondition_i + \beta_{TT} \cdot TypingScore_i + \beta_{PT} \cdot PretestScore_i + \beta_{FO} \cdot FirstOperand_i$$
$$+\beta_B \cdot Block_i + \beta_{B\times LC} \cdot Block_i \cdot LearnCondition_i + \alpha_{j[i]} + \alpha_{k[i]} + \alpha_{j[i],k[i]} + \beta_{j[i]} \cdot Block_i + \beta_{k[i]} \cdot Block_i + \beta_{j[i],k[i]} \cdot Block_i$$

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_j} & \rho_{\alpha_j\beta_j}\sigma_{\alpha_j}\sigma_{\beta_j} \\ \rho_{\alpha_j\beta_j}\sigma_{\alpha_j}\sigma_{\beta_j} & \sigma^2_{\beta_j} \end{pmatrix} \right)$$

$$\begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_k} & \rho_{\alpha_k\beta_k}\sigma_{\alpha_k}\sigma_{\beta_k} \\ \rho_{\alpha_k\beta_k}\sigma_{\alpha_k}\sigma_{\beta_k} & \sigma^2_{\beta_k} \end{pmatrix} \right)$$

$$\begin{pmatrix} \alpha_{j,k} \\ \beta_{j,k} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_{j,k}} & \rho_{\alpha_{j,k}\beta_{j,k}}\sigma_{\alpha_{j,k}}\sigma_{\beta_{j,k}} \\ \rho_{\alpha_{j,k}\beta_{j,k}}\sigma_{\alpha_{j,k}}\sigma_{\beta_{j,k}} & \sigma^2_{\beta_{j,k}} \end{pmatrix} \right)$$

(Observations, *i*, are nested within Participants, *j*, and Problems, *k*)

The above model was also run on the data from each condition separately and without the $Block \cdot LearnCondition$ and $LearnCondition$ terms to estimate the improvement by block in each of the two conditions. Arguing against a "caught off guard by the test format" explanation for

computation participant's poor performance, I find that while participants in both the memorization condition (median: 0.23, MAD: 0.09, $P(\beta_B \leq 0) = 0.01$) and the computation condition (median: 0.27, MAD: 0.14, $P(\beta_B \leq 0) = 0.04$) perform better on later speed test blocks, suggesting that participants do benefit from warming up to the task, there is no significant difference in the amount of improvement by condition (median: -0.03, MAD: 0.11; Figure 2.7; posterior distributions not shown). Notably, this analysis also rules out the possibility that computation participants are computing some answers during the test itself (perhaps by taking advantage of both the 2 second response window and the 1.5 seconds between trials), and then remembering those answers on later trials, i.e., that computation participants are continuing to learn additional answers during the test itself.

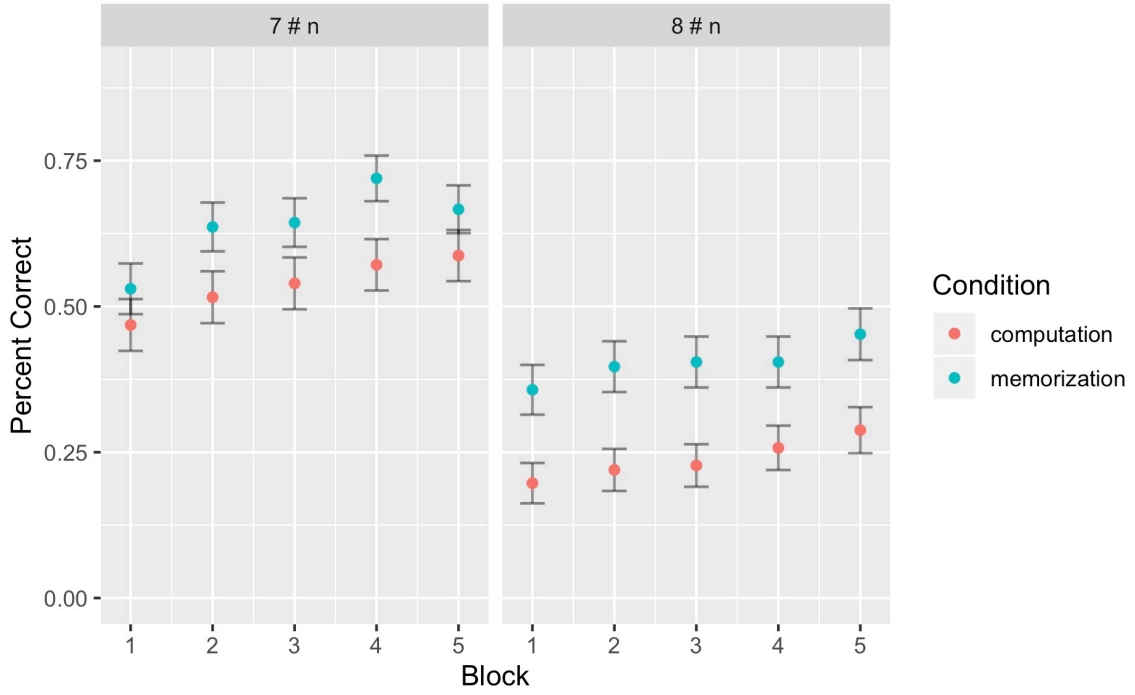

Figure 2.7. First Speed Test Performance by Block
Performance on the first speed test by block. Shown separately for participants who learned the 7 # n facts (left panel) and participants who learned the 8 # n facts (right panel) in this round. For transparency and ease of interpretation, means and error bars represent simple averages and standard errors computed from each participant's proportion correct. More precise estimates of the effect are provided by the Bayesian multi-level logistic regression model with control variables (see text), although the overall pattern of results is the same.

Incorrect Responses During Training

Participants in the computation condition were required to re-attempt a question when incorrect, while participants in the memorization condition were simply shown the correct answer. I worried that this might result in computation participants seeing more wrong answers than memorization participants. To check whether this was the case, I examined the number of wrong answers participants gave in each condition. On average, memorization participants gave incorrect answers on 26.4% of all training trials (SD: 13.2%). (Note that as my focus was on potential interference from incorrect answers, here I did not count as "incorrect answers" those training trials on which a participant entered "n", indicating that they did not know the answer.) This means that on average memorization participants saw zero wrong answers prior to seeing the correct answer on 73.6% of trials and saw one wrong answer on 26.4% of trials. On average, computation participants saw zero wrong answers (i.e., answered correctly on the first attempt) on 87.5% of trials, saw one wrong answer on 9.2% of trials, saw two wrong answers on 2.4% of trials, and saw three or more wrong answers on only 1.2% of trials. Overall, then, participants in the computation condition did not expose themselves to many wrong answers before getting a problem correct; they rarely answered a problem incorrectly more than once (3.6% of trials). Furthermore, the sum total of their incorrect answers, i.e., counting each of the multiple wrong answers to a problem separately, was actually less than the total number of incorrect answers given by the memorization participants (739 incorrect answers given by computation participants versus 1093 given by memorization participants across 4128 trials in each condition).

Training Time

As expected, computation participants spent considerably longer on training than memorization participants. It took computation participants an average of 10.2 minutes (sd = 4.6)

to complete the first training, compared with only 5.9 minutes (sd = 1.5) for the memorization participants. A similar difference was observed on the second training: 9.6 minutes (sd = 4.3) versus 5.3 minutes (sd = 1.0). The fact that the computation participants spent longer on the second training also means that they had a longer delay between their first speed test and their third speed test (retesting the first set of facts).

2.5. Discussion

In Study 1, participants learned artificial arithmetic facts via either flashcard-like memorization or self-computation. Participants learned one set of six facts, took a speeded test on them, learned a second set of facts, took a speeded test on those, and finally were retested on the first set of facts. Participants also completed a fourth speed test that intermixed novel items. This was included as a control measure, testing the validity of my assumption that participants were recalling, not computing, answers on the speed tests.

Memorization participants were able to recall more of the facts than computation participants on all three speed tests. Results from the fourth speed test, testing novel items, indicate that, as intended, the speed tests largely measure recall ability, not computation ability. However, computation participants may be computing the answers, rather than directly recalling them, on a small fraction of speed test trials, making the true advantage of memorization over computation perhaps even greater than the one I report here. Overall, these results suggest that if self-computation provides any mnemonic advantage, it is not due to directly strengthening the memory trace. That is, computing the answer oneself does not appear to somehow make the memory stronger or easier to later retrieve. Indeed, if anything, self-computation makes it more difficult to later rapidly recall the answers.

The present experiment cannot determine the reason for this disadvantage, but rather only that it exists. I speculate that it may be largely driven by attention. That is, participants in the computation condition may be so busy focusing on executing the computation that they spend very

little time attending to the final answer itself, whereas memorization participants focus exclusively on the final answer. Of course other possible mechanisms exist, and there is likely not a single driver of this difference. For example, it is possible that the retrieval practice that memorization participants engage in is a particularly strong driver of memory, either because it directly strengthens the memory trace, or because conscious mnemonic devices or less explicit mental elaboration later assist memorization participants in recalling the answers (see discussion of Bradshaw and Anderson's, "network redundancy" versus "inferential redundancy" above in section 1.3). Again this experiment was not designed to tease apart these theoretical mechanistic possibilities, but rather to make recommendations for real students in classrooms, learning arithmetic facts. Based on my results, I conclude that if you give a student a page of arithmetic problems to solve, in the course of solving them, they may commit some of the answers to memory, but this commission to memory is not obligatory, and they likely will recall fewer of their answers than a peer who directly studied those facts via flashcards.

A few additional results are worth commenting on. In this study, I wondered whether memorization might be more susceptible to either practice or retroactive interference, leading to better initial performance, but worse performance on subsequent tests. I did not find evidence that memorization participants suffered from more proactive or retroactive interference than their peers in the computation condition. However, my posterior estimates of this difference are quite wide, so it is worth noting that there is also no clear evidence that they *didn't* suffer from more interference. In other words, I really can't comment on the question of interference. It is also worth noting that, with regard to retroactive interference, the computation participants were at a disadvantage due to having a longer delay between the first and third speed tests (because it took them longer to complete the second training), making the comparison a bit unfair.

Finally, it is important to note that memorization practice was not only more effective but also more efficient, with participants learning more facts in significantly less time (e.g., 5.9 minutes

versus 10.2 minutes on the first speed test). Here, I equated the *number of trials* between the two conditions, but in a classroom setting, with a *fixed total amount of time* for math practice, memorization students should be able to practice more facts in the allotted time, resulting in an even larger overall advantage than the one reported here.

3. CHAPTER THREE Study 2: The Role of Self Re-Presentation

3.1 Overview

Study 1 found no benefit of practicing via computation on either immediate or delayed recall. These results are consistent with other lab-based studies that typically report either no difference between conditions or an advantage of memorization (e.g., Logan and Klapp 1991; Cerella, Onyper, and Hoyer, 2006; Pyke and Lefevre, 2011). However they contrast sharply with the general consensus in the educational psychology literature that computation practice results in better memory than rote memorization (e.g., Isaacs and Carroll, 1999; Baroody, Bajwa, and Eiland, 2009). In this study, I dig deeper, exploring a mechanism by which self-computation might result in superior long-term memory in real world contexts.

Suppose that a student has learned arithmetic facts by practicing computing the answers. For example, a student may have learned to compute 6 x n multiplication facts by recalling 5 x n and adding n one more time, e.g., 6 x 7 = 35 + 7. Suppose further that the student is completing a homework assignment or test and must answer 6 x 7 = ____, but unfortunately, they cannot recall the answer. The student would likely immediate fall back on the computation strategy that they learned, recomputing 6 x 7 as 35 + 7. Importantly, not only will this student get the problem correct on that assignment, but the act of correctly answering the question, should strengthen the association between the problem and its answer in the student's mind. This, in turn, should make the student more likely to correctly *recall* (not compute) that answer in the future. This follows directly from Siegler's (1988) proposal that each time a student answers a question correctly, the association between the problem and its answer is strengthened. Relatedly, Siegler (1988) also posits that each time a student answers incorrectly, the association between a problem and that *incorrect* answer is strengthened, competing with the correct answer, and decreasing the chance that the correct answer will be recalled in the future.

Now, by contrast, consider a student who originally learned the same facts via pure rote memorization. What happens if this student comes across 6 x 7 = ____ and cannot recall the answer? Although the student has previously learned that 6 x 7 = 42, they never really understood why that was - they just memorized it. In an extreme case, they may literally have no idea that 6 x 7 means to add 6 seven times (or to add 7 six times). (Note that although this extreme scenario is relatively unlikely, though not impossible, in the case of multiplication, where most students are at least exposed to the idea that multiplication is repeated addition, it is not so far fetched for other mathematical content, where we do often teach kids to just "memorize it" and not worry about why or how it works. For example, students may be taught the algorithm for long division, or to "flip and multiply" when dividing fractions without understanding the meaning of these procedures.) In the face of this recall failure, the student is unlikely to answer the question correctly, and unlike the memorization student, their future memory for the correct answer will be no better than it currently is. In fact, in the case where the student answers *incorrectly,* that incorrect answer may directly compete with the correct answer in memory, actively hurting their future ability to recall the answer (Siegler, 1988).

Furthermore, even in a less extreme case where the memorization student at least *knows* what multiplication represents, they may not be *practiced* at computing the answers if they have only ever focused on memorizing them. For example, they may think that the only way to solve 6 x 7 is to add seven six times, not realizing that there are faster ways (e.g., recalling that 5 x 7 = 35 and adding just one more 7). The student may decide that adding six sevens is too time consuming or difficult to attempt, preferring to simply enter their best guess for 6 x 7, which may be incorrect. Or if they are familiar with the 6 x n = 5 x n + n method, but have not really practiced it, they may forget if they are supposed to do 5 x 7 + 6 or 5 x 7 + 7 in order to find the answer to 6 x 7, resulting in an incorrect answer. In any case, we can expect that the student who has focused purely on memorizing

the answers will be less willing and/or less able to compute those answers when they are forgotten, and therefore more likely to respond incorrectly or not at all.

One other caveat, is that of course, a memorization participant might also choose to look-up a forgotten answer by pulling out the flashcards that they originally used to study or by using a reference table or calculator. Doing so should reinforce the connection between the correct answer and the problem in their memory, just as recomputing does in the computation learner's memory. However, I would predict that, overall, memorization students will look up the answer less frequently than computation students will recompute the answer. Why? First, and foremost, a great number of academic assignments do not permit students to use a calculator or table. This is true of classroom tests and quizzes, daily math "sprints", some standardized tests, and even most homework assignments - although some students may choose to ignore the instructions and use a calculator anyway. Furthermore, even when calculator or table use is not expressly forbidden, if the student does not already have the calculator or table in front of them, they may not be motivated enough to go get it, especially if there are only a few problems on the assignment that they do not know or if they don't particularly care about their score on the assignment. By contrast, a computation student doesn't need to get up and grab any external tools in order to do the computation, they need only think about it, which can be done quite rapidly, especially in the case where the computation is well-practiced. Additionally, in real world problem solving contexts, such as when trying to mentally figure out how much money they need to buy five candies while standing in a store, a memorization student may not have a calculator or look up table nearby, whereas a computation student always has their mind available.

Overall, then, *when an answer cannot be recalled*, a student who has practiced self-computation is expected to be more likely to answer the item correctly than a student who has learned via rote memorization, because they are more able and/or more willing to compute the answer. Importantly, by re-exposing themselves to the correct answer via computation, a phenomenon I'll call self-re-

presentation, computation students should improve their ability to *recall* those answers *in the future*. Put another way, the key difference is that computation practice allows for *self*-reinforcement of the correct answers, while memorization practice makes the learner perpetually dependent on *external* reinforcement. That is, without their flashcards or a calculator or table in front of them, they cannot be certain what the correct answer is. If that external reinforcement is not always available, then the memorization learner's memory for the correct answers is expected to decline over time. By contrast, self-reinforcement should cause the computation learner's memory to improve with time. In short, there is something incredibly powerful about finding oneself in the state of, "I can't recall X, but I can figure it out." Not only should the ability to "figure it out" boost performance on the present assignment or assessment, but it should further improve *future* performance by increasing the number of times that the student sees the problem and its correct answer together.

Prior research has typically not paid attention to this latent memory potential, instead measuring only what a student knows *today*. Of course, some studies assess learning at multiple time points, e.g., on a post-test and then a delayed post-test days or weeks later, but there is reason to think that such a design may not adequately capture this phenomenon. Specifically, this effect should emerge not simply as a result of the passage of time, but should appear only when learners engage in additional practice without external reinforcement during that delay. For example, in an earlier unpublished study that I conducted, participants learned teen multiplication facts (e.g., 17 x 7) via either self-computation or flash-card like memorization, completed a speeded post-test, and then returned one week later for another speeded test. Participants were not forewarned that they would be retested on the facts during the second session, so it is unlikely that many of them practiced the facts during the intervening week. I found no difference in performance on the delayed test by condition (although performance was generally quite low on that test). Similarly, in Study 1 (see Chapter 2), participants completed both an immediate and a delayed speed test on the first set of facts that they learned (i.e., speed tests 1 and 3). Memorization participants showed an advantage on

both the immediate and the delayed tests. From these results we might be tempted to conclude that computation confers neither an immediate nor a longer term benefit. However, crucially, the participants did not have an opportunity to practice the learned facts between the two tests. The latent potential conferred by self-computation practice is predicted to translate into an actual memory benefit only in the case where participants engage in additional practice *without external reinforcement*. Importantly, it is hypothesized that, in the real world, learners *will* frequently experience that type of practice without external reinforcement, e.g., on homework assignments, daily sprints, tests, etc., as outlined in the preceding paragraph. This makes the understanding of such a phenomenon not just theoretically interesting, but essential for designing effective study experiences.

Overall, then, Study 2 had two aims: (1) to replicate the primary finding from Study 1 that memorization results in better memory immediately after study, and (2) to test the hypothesis that computation participants might enjoy a longer-term memory advantage due to being able to self-re-present the correct answer when it is forgotten.

3.2. Method

This study was pre-registered on aspredicted.org (identifier #39642) on April 21st, 2020.

Participants

92 undergraduates were recruited to participate for psychology course credit. Of these, 83 completed the experiment. Of the nine participants who did not complete the experiment, only one left after assignment to a condition (memorization). Five participants who completed the experiment were excluded from our analyses (four due to technical glitches resulting in incomplete data, and one who did not complete the experiment in one sitting), yielding a final sample of 78 (38 female, 1 non-binary, mean age = 19.8 years).

Design

Study 2 was a single-session online experiment with participants randomly assigned to one of two learning conditions: computation (n = 39) or memorization (n = 39). Unlike Study 1,

participants in this experiment were *not* supervised by an experimenter[1]. Like Study 1, the first four activities in Study 2 were a typing assessment, a multiplication pre-test, a training round, and a speed test. Departing from Study 1, participants next completed an additional five blocks of practice on the same facts but this time *without feedback* (Figure 3.1). This "feedback-less practice" (which was billed to participants as an "untimed test") was included to determine how the two conditions would respond differently to forgetting the trained facts. Specifically, I predicted that, during the feedback-less practice, the memorization participants would correctly answer at most as many items as they did on the first test, but that the computation participants would answer more items correctly, due to computing answers that they could not recall. Finally, participants completed a second speed test that was identical to the first speed test. I predicted that the feedback-less practice would benefit the computation participants, whose performance would improve from the first to the second speed test, while the memorization participants would see no improvement. Since, unlike Study 1, in this study each participant learned only a single set of facts, there was no need to use two different sets of facts. I elected to have all participants learn only the 8 # n facts (which, incidentally, happened to be the harder set from Study 1, although, as noted in the footnote, I didn't yet know which set was more difficult when I designed Study 2).

---

[1] Study 2 was actually conducted prior to the Study 1 reported here. An earlier version of Study 1 was started in January of 2020 with in-person subjects, but was paused approximately halfway through data collection in March 2020 due to the global pandemic. Study 2 was run online in Spring 2020 and was designed to be a follow-up to Study 1. I initially hoped that we could resume in person testing for Study 1 when the pandemic ended, but when it became apparent that in-person research would not resume for many months, I began a new version of Study 1 online in Summer 2020. Hence, Study 1 (originally winter 2020, re-done in late summer 2020) was actually completed after Studies 2 (spring 2020) and 3 (early summer 2020). However, I present them in this order here because Studies 2 and 3 were designed as a follow-up to study 1 and the research questions make the most sense in that order.
I felt good about the quality of the data that I collected from the unsupervised online participants in Studies 2 and 3, particularly since, at the end of their session, participants were asked to self-report on the quality of the data that they provided (e.g., the effort they gave, whether or not they followed the instructions, etc.) and explicitly asked whether or not they followed all instructions (See questions in Appendix E). To encourage honesty, participants were explicitly told that these answers would not affect their compensation. However, upon further reflection I worried that by explicitly telling participants at the end that they would receive full compensation regardless of whether they followed the instructions or paid attention, I was discouraging our subjects from giving their full effort in future unsupervised online studies. As a result, in Studies 1 and 4, I supervised my participants via Zoom. Again, because the studies are reported out of chronological order here, that means that Studies 1 and 4 were supervised, while Studies 2 and 3, which took place earlier, were not.

Study 1	Study 2	Study 3	Study 4
Typing Test	Typing Test	Typing Test	*Typing Test*
Multiplication Test	Multiplication Test	Multiplication Test	*Multiplication Test*
Delay 24 h to 4 weeks			
*Practice 8s**	Practice 8s	Practice 8s	*Practice 8s*
Test 8s	Test 8s	Test 8s	*Test 8s*
Practice 7s		Practice 7s	*Practice 7s (new rule)*
Test 7s			*Test 7s*
	Untimed Test 8s	Untimed Test 8s	*Untimed Test 8s*
Test 8s	Test 8s	Test 8s	*Test 8s*
Test 7s, 8s, & 9s			

Figure 3.1. Comparison of the experimental sequence across Studies 1 through 4.
Cells appearing in the same color represent the same task (with possible slight modifications, e.g., perhaps differing in the number of trials, as described in the text). Except for the "untimed test," all "tests" were speeded, i.e., with two seconds to respond to each item. Italics denote portions of the experiment that were supervised by an experimenter via Zoom (to clarify this was the portion of Study 1 after the delay and all of Study 4). Black cells indicate where literally nothing happened (i.e., participants went directly from the task above the black cell to the one below it); they are included in the figure purely to facilitate comparisons across studies, i.e., so that identical tasks appear in the same row.
* In Study 1, only, half of the participants in each condition learned the facts in the opposite order, i.e., for those participants, every entry that reads "8s" was actually"7s" and every entry that reads "7s" was actually "8s".

Materials

Materials and apparatus were as described in Study 1.

Procedure

Typing and Multiplication Tests. The typing assessment itself was identical to Study 1. However, the cut off for passing the typing assessment was much lower in this study (20% versus 70% in Study 1). See Appendix F for an explanation of these exclusion criteria as well as for a re-analysis of the Study 2 data excluding participants who scored below 70% on the typing measure

(i.e. for a sample comparable to Study 1). The overall results remain unchanged when this more restrictive criterion is applied.

The multiplication pre-test was similar to Study 1 but included all single-digit multiplication combinations of the digits 2 through 9 in both commuted orders, whereas Study 1 excluded x2 and x5 facts. (There was less pressure here to keep the pre-test short in this experiment because it was not a separate paid screener; see Appendix C for a complete list of multiplication test items.) Like the typing measure, the threshold to pass the multiplication test in this study was lower (20%) than in Study 1 (40%). Again, see Appendix F for a reanalysis with more restrictive exclusion criteria.

Training and First Speed Test. The first training was identical to Study 1 with the exception that the training consisted of 10 blocks of practice (instead of 8) for a total of 60 trials, and all participants learned 8 # n facts. The first speed test was identical to Study 1.

Feedback-less Practice. Immediately after the first speed test, participants saw brief instructions introducing the feedback-less practice (new for Study 2). Although I thought of this part of the experiment as feedback-less practice, it was billed to the participants as an "untimed test" both to help them understand the format and to encourage them to give their best effort. Practically, this segment of the experiment was the same as the speed test except with unlimited time to respond to each problem. Like the speed tests, the untimed test did not differ by condition.

Second Speed Test. A second speed test, identical to the first, followed the untimed test (feedback-less practice).

Follow-Up Questions. As in Study 1, at the end of the experiment, participants completed a series of follow-up questions (Appendix E). These included the same three questions as Study 1, as well as questions that asked participants what the instructions were and whether they had followed them (since they were unsupervised). The follow-up questions also asked participants to rate the quality of the data they provided on a four point scale (poor, fair, good, excellent).

3.3 Results

Follow Up Questions

One participant failed the instructions check question and was excluded from all analyses as noted above. Eight participants self-reported providing "fair" quality data, and none reported "poor" quality data. Analyses without the eight "fair" participants are reported in Appendix F. The overall findings remain unchanged.

First Speed Test

Data from the first speed test were analyzed using the same model described in Study 1, with the exception that the $\beta_{FO} \cdot FirstOperand_i$ term was dropped because everyone learned 8 # n problems in this study. Replicating the results from Study 1, on the first speed test, participants in the memorization condition outperformed participants in the computation condition (56% correct versus 40% correct; PPE: median = 1.04, mad = 0.33, $P(\beta_{LC} \leq 0) = 0.001$; Figures 3.2 and 3.3)

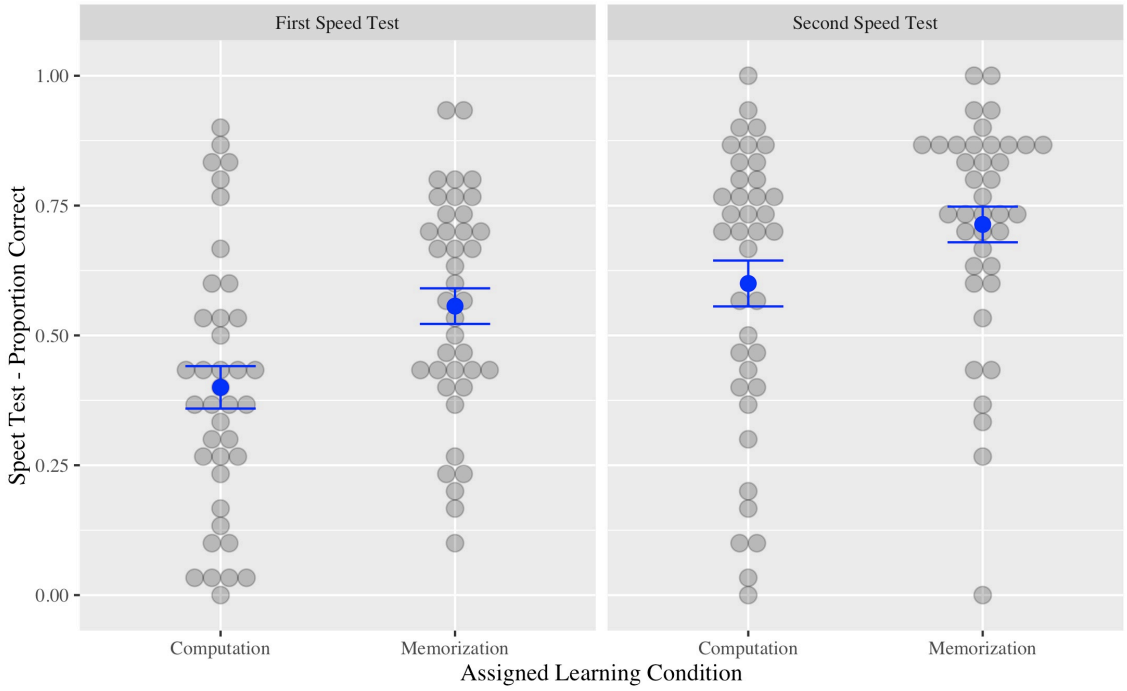

Figure 3.2. Performance on the two speed tests by assigned learning condition.
For transparency and ease of interpretation, blue dots and error bars represent simple averages and standard errors of the displayed data points (i.e., each participant's proportion correct). More precise estimates of the effect are provided by the Bayesian multi-level logistic regression model with control variables (see Figure 3.3), although the overall pattern of results is the same.

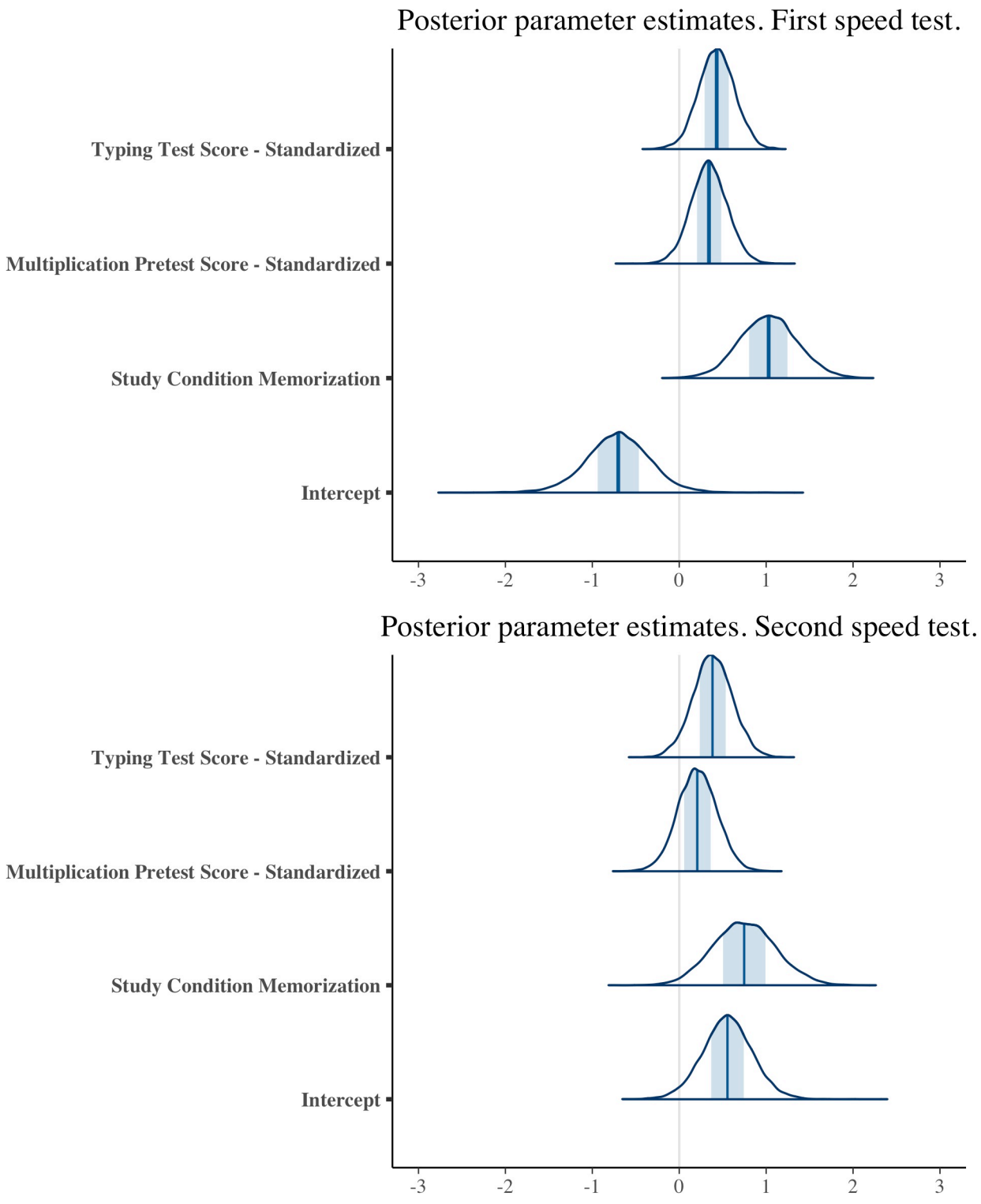

Figure 3.3. Posterior parameter estimates for the models fit separately to the first and second speed test data.
Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Study Condition Memorization".

Second Speed Test

Data from the second speed test were analyzed using the same model as was used for the first speed test. On the second speed test, memorization participants continued to perform better than computation participants (71.3% versus 60.0% correct; PPE: median = 0.75, mad = 0.36, $P(\beta_{LC} \leq 0) = 0.02$; Figure 3.3).

First versus Second Speed Test

Data from the first and second speed tests were compared using the following model (different from that used in Study 1 due to the fact that everyone learned the same set of facts here):

$$logit(p_i) = \alpha_0 + \beta_{LC} \cdot LearnCondition_i + \beta_{TT} \cdot TypingScore_i + \beta_{PT} \cdot PretestScore_i + \beta_{TR} \cdot TestRound_i$$
$$+\beta_{TR\times LC} \cdot TestRound_i \cdot LearnCondition_i + \alpha_{j[i]} + \alpha_{k[i]} + \alpha_{j[i],k[i]} + \beta_{j[i]} \cdot TestRound_i + \beta_{k[i]} \cdot TestRound_i + \beta_{j[i],k[i]} \cdot TestRound_i$$

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_j} & \rho_{\alpha_j\beta_j}\sigma_{\alpha_j}\sigma_{\beta_j} \\ \rho_{\alpha_j\beta_j}\sigma_{\alpha_j}\sigma_{\beta_j} & \sigma^2_{\beta_j} \end{pmatrix} \right)$$

$$\begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_k} & \rho_{\alpha_k\beta_k}\sigma_{\alpha_k}\sigma_{\beta_k} \\ \rho_{\alpha_k\beta_k}\sigma_{\alpha_k}\sigma_{\beta_k} & \sigma^2_{\beta_k} \end{pmatrix} \right)$$

$$\begin{pmatrix} \alpha_{j,k} \\ \beta_{j,k} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_{j,k}} & \rho_{\alpha_{j,k}\beta_{j,k}}\sigma_{\alpha_{j,k}}\sigma_{\beta_{j,k}} \\ \rho_{\alpha_{j,k}\beta_{j,k}}\sigma_{\alpha_{j,k}}\sigma_{\beta_{j,k}} & \sigma^2_{\beta_{j,k}} \end{pmatrix} \right)$$

(Observations, *i*, are nested within Participants, *j*, and Problems, *k*)

Additionally, to analyze the amount of improvement in each condition, a modified version of this model without the two "LearnCondition" terms was run separately on the data from each condition.

Participants in both conditions improved from the first to the second speed test (Figure 3.4); on average computation participants improved by 20.0 percentage points and memorization participants improved by 15.7 percentage points (PPE: Computation - median = 1.26, mad = 0.19, $P(\beta_{TR} < 0) < 0.0001$; Memorization - median = 0.99, mad = 0.23, $P(\beta_{TR} < 0) = 0.0006$). There was not a clear *difference* by study condition in the amount of improvement (PPE: median = -0.27, mad = 0.24, $P(\beta_{LC\times TR} > 0) = 0.13$; Figures 3.4 and 3.5), although it trended towards the memorization condition showing less improvement than the computation condition.

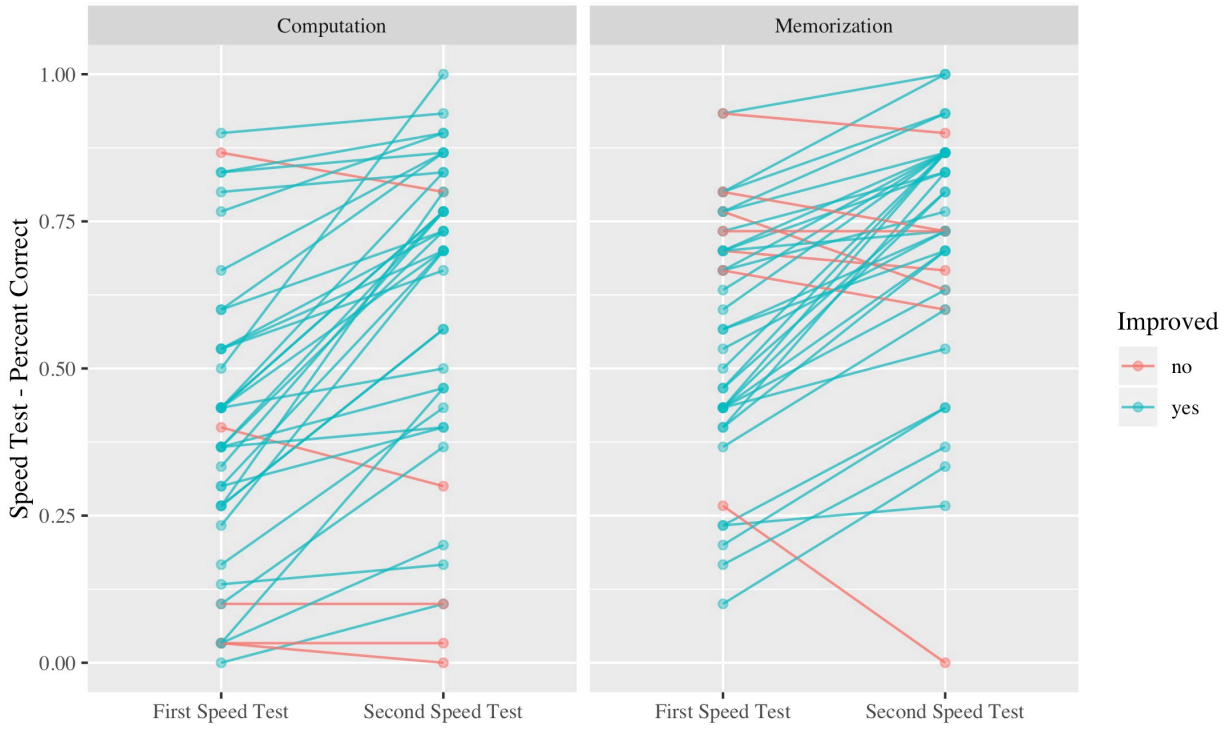

Figure 3.4. Change in performance from the first to the second speed test.
Lines connect the two test scores belonging to the same participant. Teal colored lines are used for participants who improved from the first to the second speed test, red lines for those who did not improve.

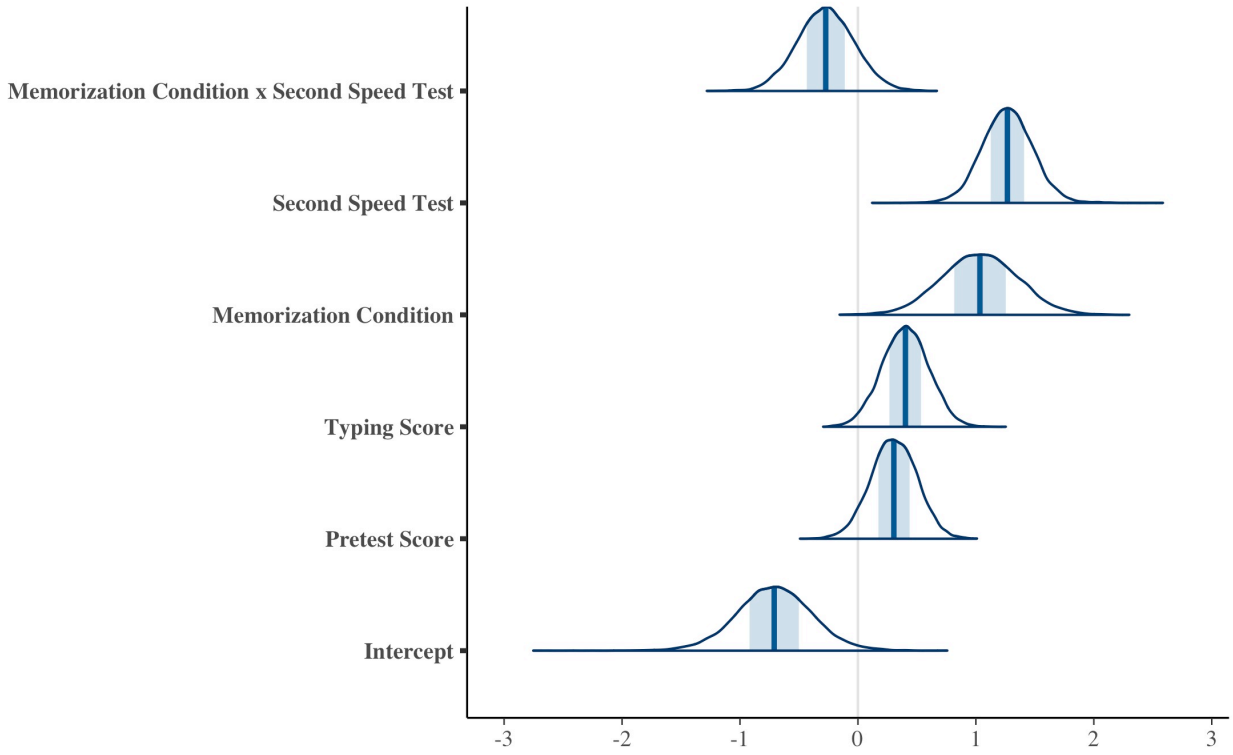

Figure 3.5. Posterior parameter estimates for the model comparing the first and second speed test data.
Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Memorization Condition x Second Speed Test".

Untimed Test

Performance on the untimed test was analyzed using the same model used to analyze the first speed test data. Participants in both conditions performed very well on the untimed test (Computation: 93.5% correct; Memorization: 92.6% correct, Figure 3.6). Performance on the untimed test did not differ by condition (PPE: Median = 0.10, MAD = 0.55, $P(\beta_{LC} \leq 0) = 0.43$, Figure 3.7).

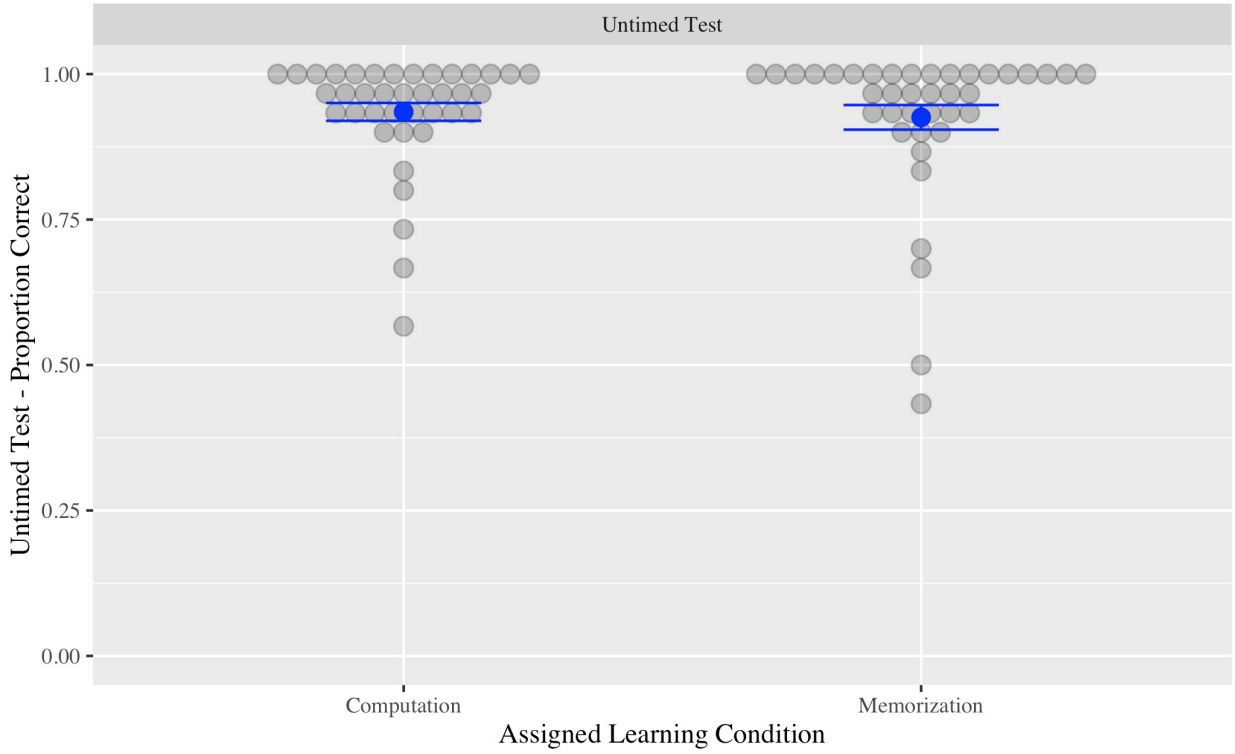

Figure 3.6. Performance on the untimed test by assigned learning condition.
For transparency and ease of interpretation, blue dots and error bars represent simple averages and standard errors of the displayed data points (i.e., each participant's proportion correct). More precise estimates of the effect are provided by the Bayesian multi-level logistic regression model with control variables (see Figure 3.7), although the overall pattern of results is the same.

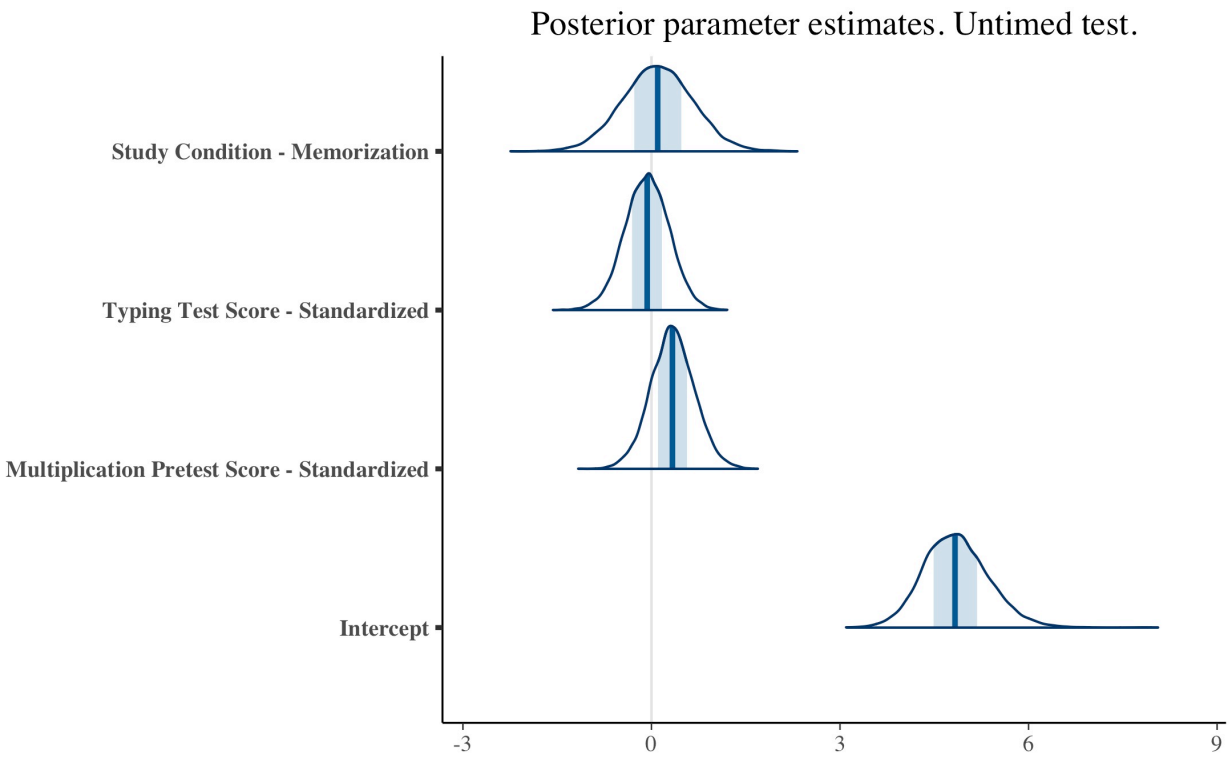

Figure 3.7. Posterior parameter estimates for the model of the untimed test data. Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Study Condition - Memorization".

Recall that each problem was presented five times on the first speed test. I separately analyzed only those problems that a participant answered incorrectly on all five presentations on the first speed test, i.e., that I would have concluded that a participant definitely did not "know" as of the first speed test. This was a reasonably sized subset of the data; out of 468 items (6 different problems times 78 participants), 111 items (71 computation, 40 memorization) fell into this category. Despite never answering them correctly on the first speed test, participants in both conditions were able to correctly answer many of these items on the untimed test (Computation: 87.0% correct, Memorization: 71.5% correct). Memorization participants correctly answered fewer of these items than computation participants (PPE: Median = -1.56, MAD = 0.89, $P(\beta_{LC} > 0) = 0.04$; posterior distributions not shown).

3.4 Discussion

In this study, participants learned a set of six artificial arithmetic facts via either flashcard-like memorization or self-computation, took a speed test on those facts, completed an untimed test (i.e., additional practice without feedback) on those facts, and then took a second speed test, again on the same set of facts. The first speed test simply measured how well participants could recall the arithmetic facts after engaging in one of two different types of practice: self-computation and flashcard-like memorization, and essentially served as a replication of Study 1. Results from that speed test were consistent with the findings from Study 1: immediately after study, participants who practice via flashcard-like memorization are better able to recall the studied facts than participants who practice via self-computation. Specifically, here the observed condition difference was 56% correct (memorization) versus 40% correct (computation).

The second speed test was designed to answer a new question. Specifically, I hypothesized that the intervening "untimed test" or "feedback-less practice" would benefit computation participants more than memorization participants, causing computation participants to improve from the first to the second speed test while memorization participants' performance stayed steady or declined. My reasoning was that if a memorization participant didn't know the answer to an item on the first speed test, they would also be unable to retrieve that answer on the untimed test, resulting in little to no improvement from the first to the second speed test. By contrast, I expected that computation participants would be able to re-compute the answers to forgotten items on the untimed test, and that this re-exposure to the correct answer would improve their performance on the second speed test.

Contrary to my predictions, I found that participants in both conditions improved substantially from the first to the second speed test. Why? Specifically, why do memorization participants improve from the first to the second speed test when I expected that the intervening feedback-less practice would be of no use to them? Some of their improvement may result from

increased familiarity with the test format. Indeed, in Study 1, participants performed better on the second speed test despite the fact that, in that experiment, the second speed test targeted a different set of facts. However it should be noted that they did not improve nearly as much in Study 1 as they improved in this study (6.1 percentage point improvement in Study 1 versus 15.7 in this study). Although some of the improvement may be attributable to task familiarity, there is reason to believe that a portion of the memorization participants' improvement may in fact result from their experience with the feedback-less practice. I had expected that any problem not successfully retrieved on the first speed test would also experience retrieval failure during the feedback-less practice, resulting in no additional learning. Contrary to this expectation, memorization participants successfully retrieved many items during the untimed practice that they had failed to retrieve on the first speed test. Indeed, while they answered only 56% of questions correctly on the first speed test, they answered 93% of questions correctly on the untimed test. Moreover, when I focused exclusively on items that were not correctly answered on *any* of their five presentations on the first speed test, i.e., that we would have thought that participants most certainly did *not* know, I found that memorization participants answered these items correctly on the untimed test 78% of the time, indicating that they did in fact "know" many of them. In retrospect, it makes sense that, on the first speed test, a memorization participant might have known the answer to a given problem but not well enough to respond in two seconds. When subsequently given unlimited time to respond during the feedback-less practice, the participant may have been able to successfully retrieve the answer, and that successful retrieval may, in turn, have strengthened the participant's memory for that fact to the point where they could subsequently retrieve it in two seconds on the second speed test.

In short, my proposed mechanism hinged on there being a difference in the number of items successfully answered during this feedback-less practice, which would in turn drive a difference in performance on the final speed test. In actuality, memorization and computation participants correctly answered comparable numbers of items during the feedback-less practice (92.5% v. 93.6%

correct, respectively), and many more items than they answered on the first speed test (56% and 40% correct, respectively) which may explain why both groups benefited from the feedback-less practice, answering significantly more items correctly on the second speed test than they did on the first (71% correct and 60% correct, respectively), and with memorization continuing to significantly outperform computation participants.

Importantly, this design was likely not an ideal model for the real world phenomenon that I had hoped to model. In this experiment, the feedback-less practice occurred immediately following the initial practice, which meant that there was little time for participants to forget any of the facts that they had learned. However, in real life, hours, days, or even weeks may elapse between when a student studies with their flashcards and when they next need to recall those facts (e.g., on a homework assignment or test). Therefore, in the real world, more forgetting should occur, and more retrieval failures should result during the feedback-less practice, rendering the feedback-less practice less useful for the memorization participants. Study 3 explored this possibility.

4. CHAPTER FOUR Study 3: An Additional Test of the Self-Re-Presentation Hypothesis

4.1 Overview

Study 3 attempted to test the same hypothesis as Study 2: that flashcard-like memorization practice would be better than computation practice in the short term (i.e., immediately after practicing with the flashcards), but that computation practice would prove superior in the long term (after additional practice without the flashcards present) due to the effects of self-re-presentation. Contrary to expectation, this effect did not appear in Study 2, likely because memorization participants were almost universally successful in retrieving the correct answer during the feedback-less practice. Study 3 attempted to decrease the likelihood of successful retrieval during the feedback-less practice in two ways. First, I reduced the amount of initial practice during the learning phase from 10 blocks to 8 blocks. Second, I introduced a distractor task between the first speed test and the feedback-less practice to promote forgetting. This delay more closely mimics what occurs in real educational contexts, where students typically do not study arithmetic facts immediately prior to completing assignments or assessments that require their use.

4.2 Method

Participants

85 self-described undergraduates from the US and the UK were recruited via Prolific (prolific.co) and were offered $10 USD for completing the experiment. Of these, I prevented 16 from continuing due to poor performance on the typing warm-up (<50% correct) or the single-digit multiplication pre-test (<40% correct). An additional 5 participants voluntarily withdrew before the end of the experiment (3 after assignment to condition, 2 from the computation condition). 64 participants completed the experiment. Three participants, all from the computation condition, were excluded from further analysis on the basis of their answers to the follow up questions (e.g., indicating that they cheated by using a calculator or scratch paper). This left a final sample of 61 (27 female, 1 non-binary, mean age = 20.9 years). As a side note, I recognize that there was somewhat

greater attrition and/or exclusion from the computation condition, likely reflecting the fact that that condition was more difficult, making participants in that condition more likely to quit and/or cheat (and therefore be excluded). While this may somewhat bias the results in this study, Study 4 (see Chapter 5) replicates all of Study 3's findings perfectly and had no attrition or exclusion, making me confident that the results that I report here are not solely due to differential attrition.

Design

Study 3 was a single-session online experiment with participants randomly assigned to computation (n = 28) or memorization (n = 33). Study 3 was identical to Study 2 except that the first training was shortened from 10 blocks to 8 blocks, and an additional round of training on a new set of problems was inserted as a distractor task after the first speed test and before the feedback-less practice. The problems used in the distractor task were generated via the same algorithm, but all had 7 as their first operand (7 # 2, 7 # 4, 7 # 5, 7 # 6, 7 # 7, 7 # 9). Both of these changes (the shortened training, and the additional distractor task) were designed to increase the amount of forgetting between the first speed test and the untimed test (feedback-less practice).

Materials and Apparatus

Materials and apparatus were identical to those described for Experiments 1 and 2.

Procedure

The procedure was identical to Study 2 except for the changes necessarily implied by the changes to the design described above. For the new piece in this experiment, i.e., the distractor task / second training, participants were simply told that now they would be learning a new set of artificial arithmetic problems and were briefly reminded of the training instructions for their assigned condition. Further, for the untimed test, the instructions were identical to those given in Study 2, except the addition of a single line that noted that this untimed test would be about the *first* set of problems that participants had practiced (not the distractor set). The follow-up questions at

the end of the experiment were identical to Study 2 except for the addition of a single item that explicitly asked participants whether or not they had followed the instructions (see Appendix E).

4.3 Results

Follow-Up Questions

One participant in the memorization condition figured out the underlying algorithm. They were still included in these analyses for the reasons outlined in Chapter 2. Four participants reported providing "fair" quality data. Analyses without these four participants are reported in Appendix F, but the overall results are unchanged.

First Speed Test

Data from all tests in this study were analyzed using the same models described in Study 2. Replicating Studies 1 and 2, participants in the memorization condition performed better on the first speed test than participants in the computation condition (41.9% correct versus 31.3% correct; PPE: Median = 0.92, MAD = 0.48, $P(B_{LC} < 0) = 0.03$; Figures 4.1 and 4.2).

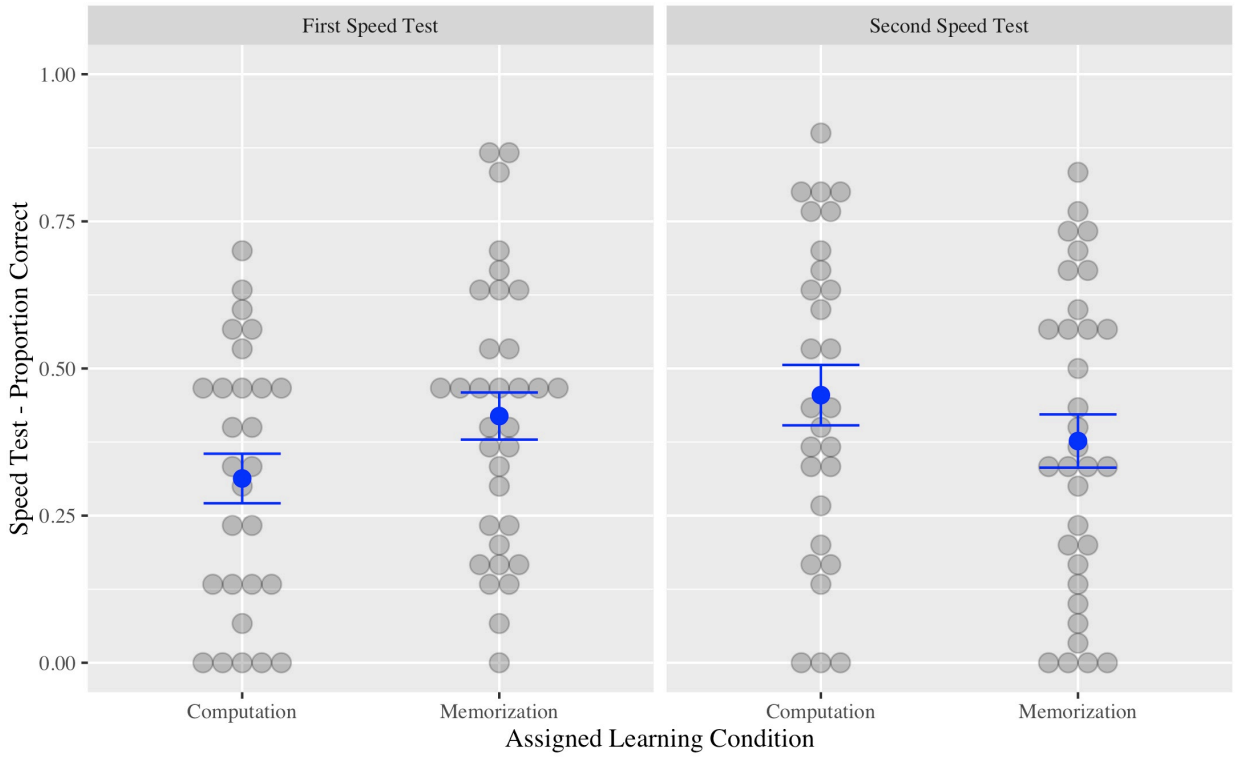

Figure 4.1. Performance on the first and second speed tests by assigned learning condition. For transparency and ease of interpretation, blue dots and error bars represent simple averages and standard errors of the displayed data points (i.e., each participant's proportion correct). More precise estimates of the effect are provided by the Bayesian multi-level logistic regression model with control variables (see Figure 4.2), although the overall pattern of results is the same.

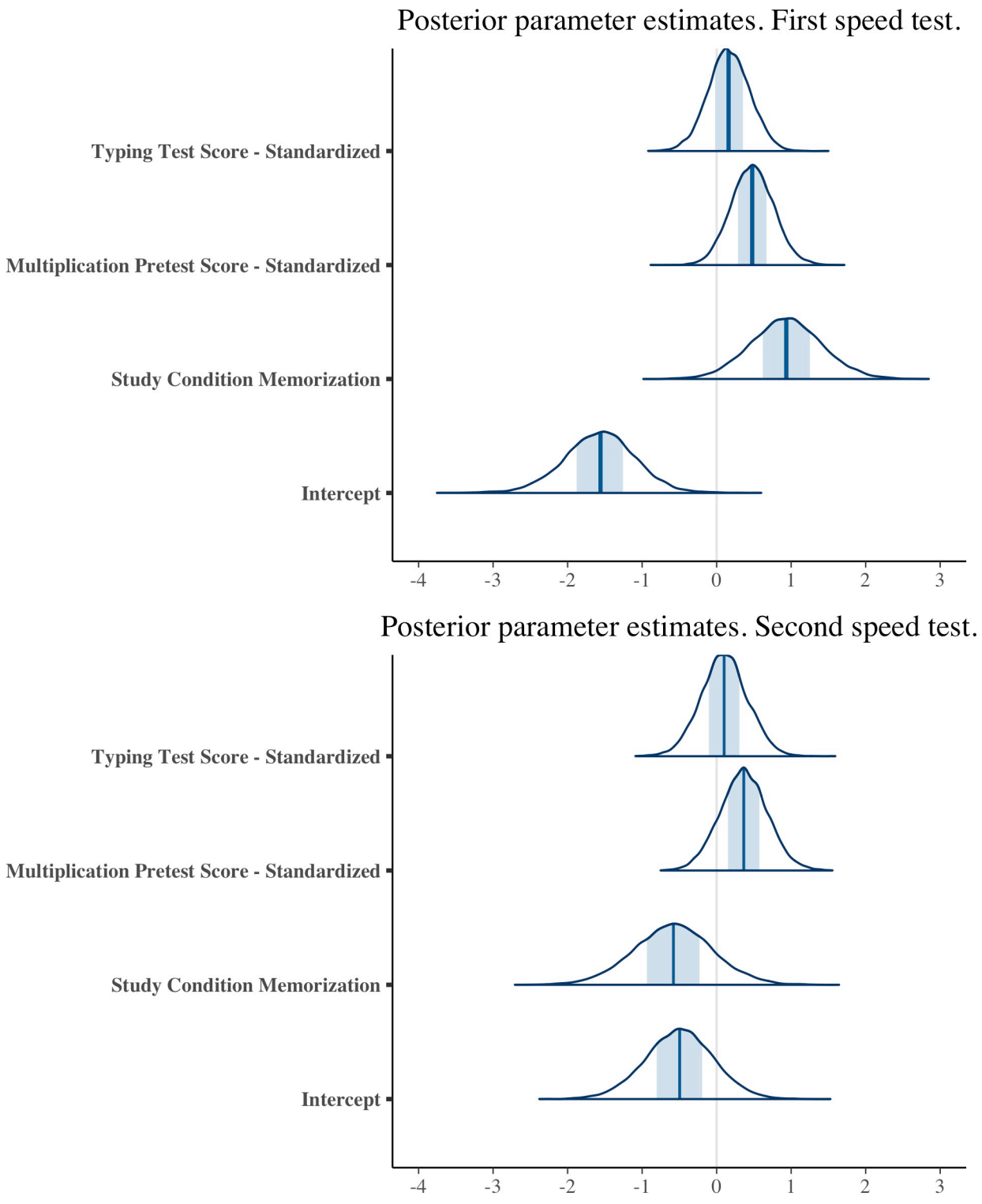

Figure 4.2. Posterior parameter estimates for the models fit separately to the first and second speed test data.
Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Study Condition Memorization"

Second Speed Test

On the second speed test, re-testing the 8 # n facts after both the distractor task and the feedback-less practice, computation participants numerically outperformed memorization participants (45.4% correct versus 37.7% correct; Figure 4.1). This is notable because it is the only speed test seen thus far (across six speed tests: three speed tests from Study 1, two speed tests from Study 2, and one earlier speed test in the present study) in which memorization participants did not answer more items correctly than computation participants. Statistically, there isn't enough evidence to say that this difference clearly favors the computation participants (PPE: Median = -0.60, MAD = 0.52, $P(B_{LC} > 0) = 0.13$, Figure 4.2), but, as discussed below, there is a clear, statistically reliable condition difference in how much participants improved from the first to the second speed test.

First versus Second Speed Test

Computation participants were able to correctly answer more items on the second speed test than the first (average change = +14.2 percentage points; PPE: median = 1.02, mad = 0.29, $P(\beta_{LC} < 0) = 0.002$; posterior distributions not shown), while memorization participants answered fewer items correctly (average change = - 4.2 percentage points; PPE: median = -0.51, mad = 0.31, $P(\beta_{LC} > 0) = 0.05$; posterior distributions not shown). The average slope (i.e., the change from speed test 1 to speed test 2) in the memorization condition was reliably more negative than the average slope computation condition (PPE: median = -1.44, mad = 0.34, $P(\beta_{LC\times TR} > 0) < 0.0001$; Figures 4.3 and 4.4).

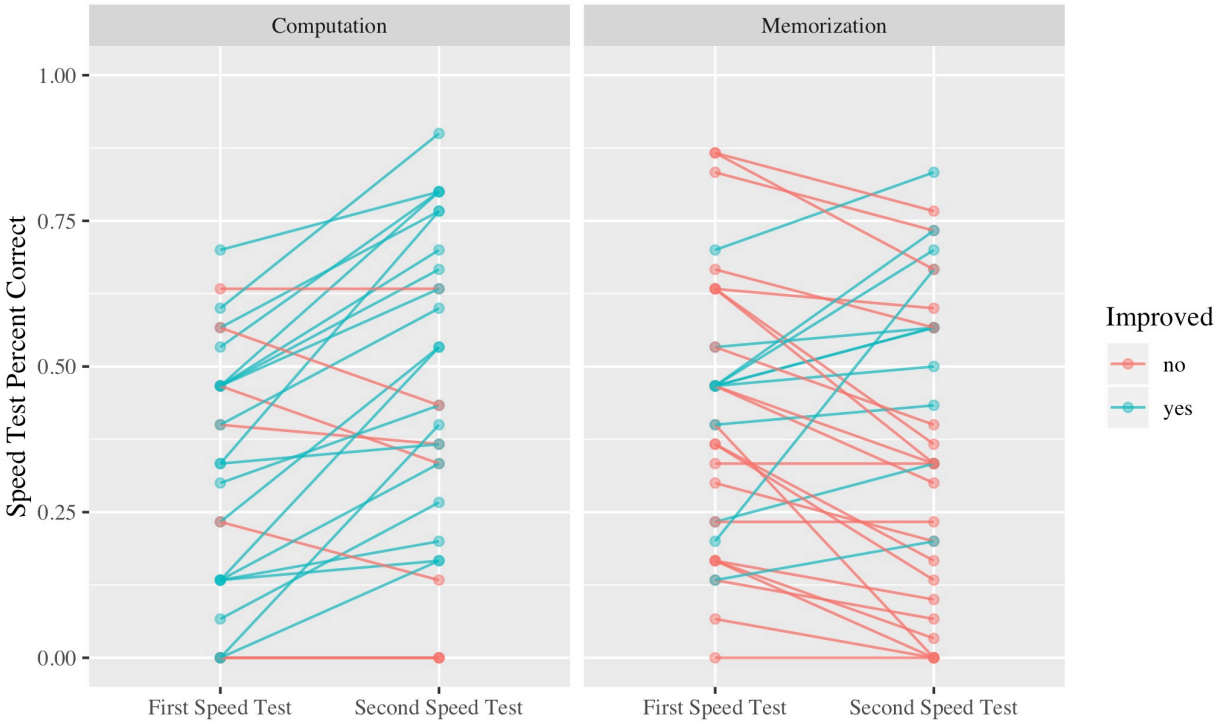

Figure 4.3 Change in performance from the first to the second speed test.
Lines connect the two test scores belonging to the same participant. Teal colored lines are used for participants who improved from the first to the second speed test, red lines for those who did not improve.

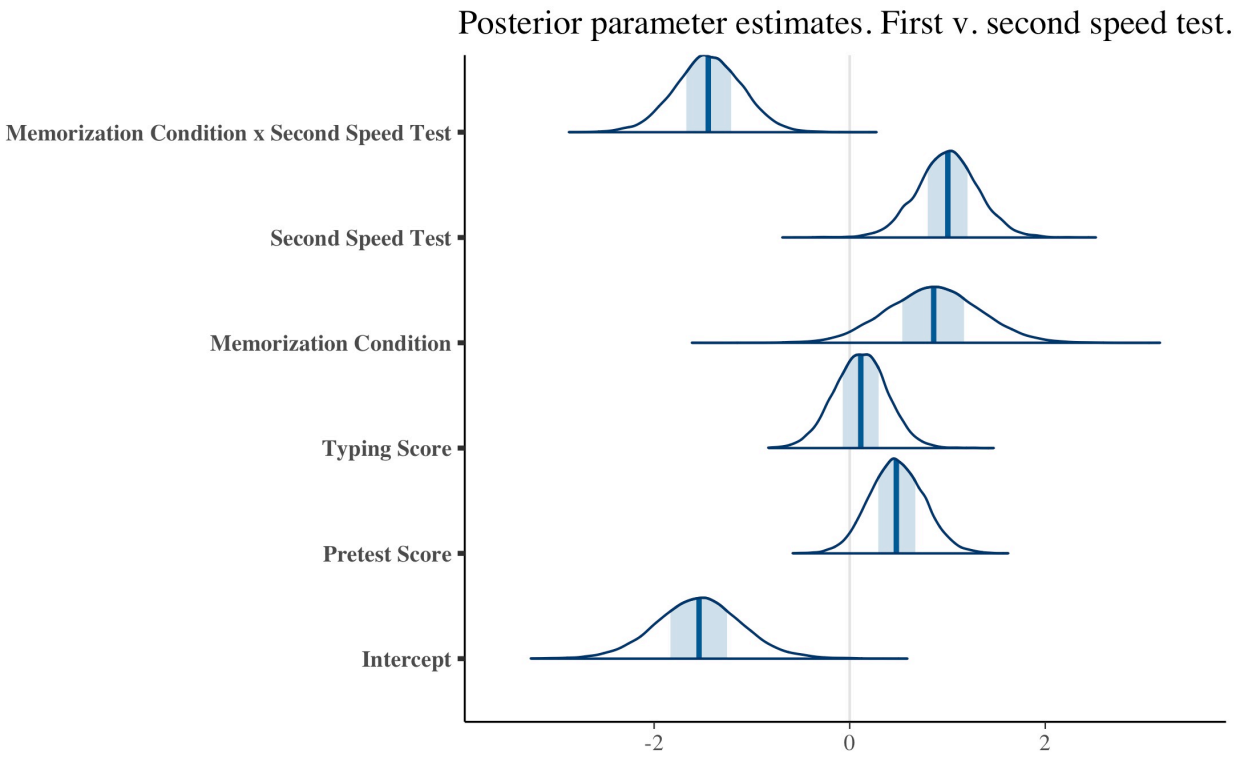

Figure 4.4. Posterior parameter estimates for the model comparing the first and second speed test data.
Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Memorization Condition x Second Speed Test".

Untimed Test

Memorization participants answered significantly fewer items correctly on the untimed test as compared to computation participants (55.4% correct versus 87.3% correct; PPE: Median = -3.35, MAD = 0.80, $P(\beta_{LC} > 0) < 0.0001$, Figures 4.5 and 4.6).

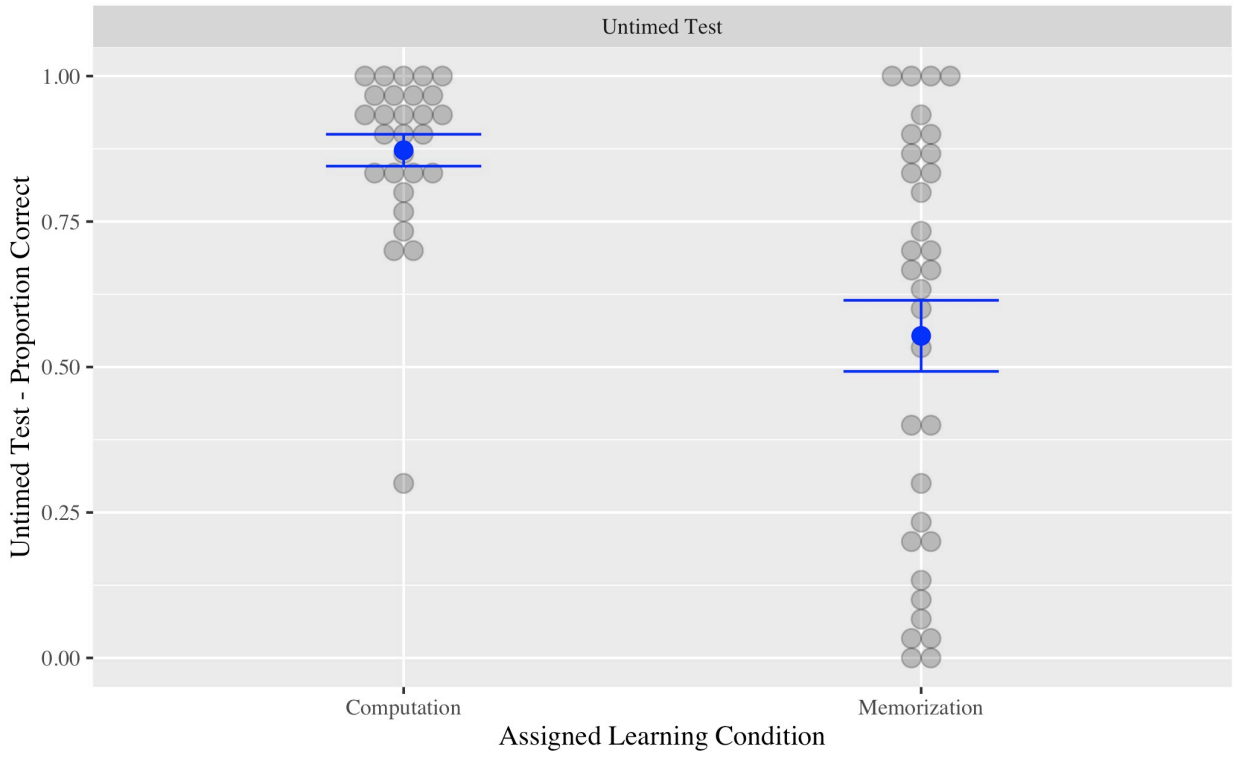

Figure 4.5. Performance on the untimed test by assigned learning condition.
For transparency and ease of interpretation, blue dots and error bars represent simple averages and standard errors of the displayed data points (i.e., each participant's proportion correct). More precise estimates of the effect are provided by the Bayesian multi-level logistic regression model with control variables (see Figure 4.6), although the overall pattern of results is the same.

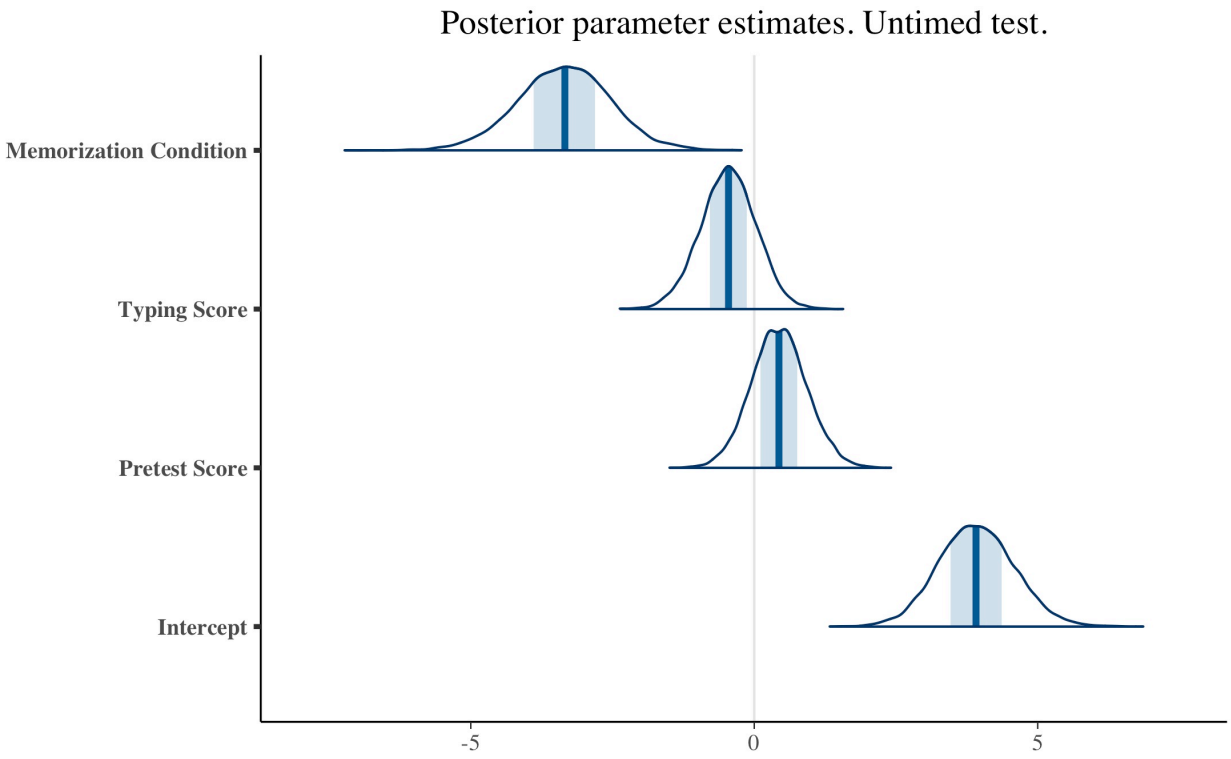

Figure 4.6. Posterior parameter estimates for the untimed test data.
Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Memorization Condition".

Additionally, I noted that memorization participants responded more quickly to items on the untimed test than did computation participants (PPE: Median = -2.23, MAD = 0.61, $P(\beta_{LC} > 0) < 0.0001$). Among items answered correctly, memorization participants responded in 2.4 seconds on average (sd = 2.0), while computation participants responded in 4.6 seconds on average (sd = 5.4) (Figure 4.7). As computation is known to take longer than recall, this supports the supposition that computation participants are computing some of their responses on the untimed test. Note that for the purposes of computing the mean response time and running the response times model reported above, a single outlier observation was removed from the memorization condition - it had a response time of 171 seconds while every single other memorization response (including the other twenty-nine responses made by that same participant) had a response time of 23 seconds or less. Even in the computation condition, which tended to have longer response times,

the longest recorded response times was only 67 seconds. Clearly this 171 second trial represented an anomaly.

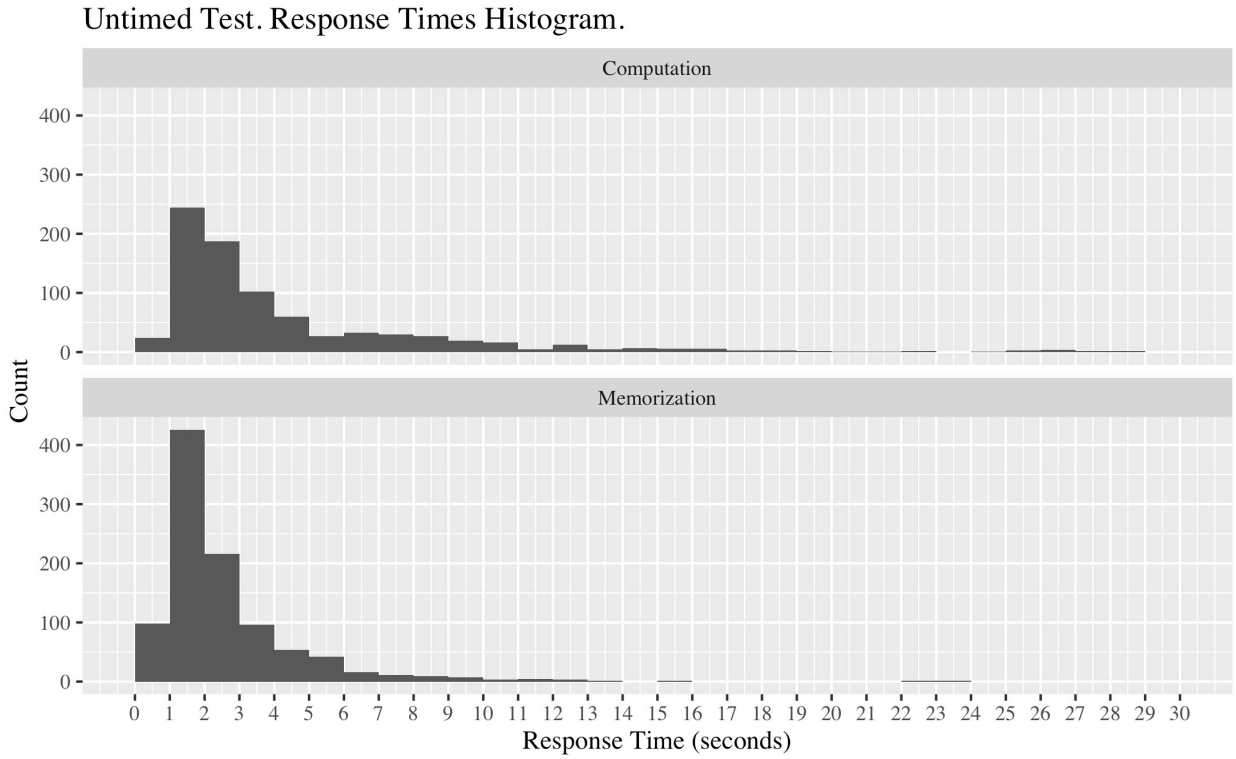

Figure 4.7. Histogram of response times for untimed test items by condition.
For the sake of readability, an additional 7 observations with response times greater than 30 seconds were not included in this plot. Six of these excluded observations were in the computation condition: 32, 34, 35, 39, 48, and 67 seconds. One was in the memorization condition: 171 seconds. As noted in the text, this 171 second response was treated as an anomalous outlier and was also excluded from my statistical analyses.

4.4 Discussion

In this experiment, the effect that I had predicted appeared: computation participants successfully answered more items on the second speed test than they had on the first (improving from 31% correct to 45% correct), while the opposite was true of the memorization participants (who declined from 42% correct to 38% correct). As expected, this effect appeared to be driven by computation participants successfully answering many more items on the untimed test than

memorization participants (87% v. 55% correct). This stands in contrast to Study 2, where participants in both conditions successfully answered nearly all items on the untimed test (93% correct in each condition), and both subsequently improved from the first to the second speed test. That is, due to the shortened training (8 blocks versus 10) and the intervening distractor task in the present study, participants in both conditions in this experiment were likely unable to directly recall many of the trained facts on the untimed test. In the face of these recall failures, memorization participants had little recourse, and answered relatively few untimed test items correctly. By contrast, computation participants could re-compute the answers to any problems they had forgotten, resulting in a high rate of success on the untimed test. Supporting this explanation, computation participants took longer to respond to the untimed test items on average (mean 4.6 seconds) than did memorization participants (mean 2.4 seconds), as would be expected if the computation group was engaging in computation on many problems while the memorization group engaged primarily in direct recall. Crucially, this difference in untimed test performance then translated into improved performance on the second speed test for computation participants, while the opposite was true for the memorization participants. Overall, the results of this study support the idea that computation may confer a long term advantage in that participants can recompute forgotten answers, which boosts their own subsequent memory for those answers.

5. CHAPTER FIVE Study 4: A More Robust Test of the Self-Re-Presentation Hypothesis

5.1 Overview

Study 4 was designed to replicate Study 3 with one important change. In Study 3, the distractor (7 # n) problems employed the same algorithm as the target (8 # n) problems. This made the memory task easier for the computation participants; when they reached the untimed test, they only needed, at a minimum, to remember the algorithm, which they had just recently practiced with the distractor problems, while the memorization participants needed to remember the six 8 # n answers, which they had not seen since the first training. To address this issue, in Study 4, the distractor set of problems (7 ~ 4, 7 ~ 5, 7 ~ 6, 7 ~ 7, 7 ~ 8, 7 ~ 9) used a different algorithm: a ~ b = 30b - (a + b), e.g., 7 ~ 4 = 109. (Recall that the original algorithm was a # b = 10(a + b) + ab). This meant that when they reached the untimed test, both groups would be forced to rely on their memory of something that they had not seen since the first training round. Overall, this served as a more robust test of my self-re-presentation hypothesis. This study also used a larger and different sample from Study 3 (undergrads participating for course credit, rather than Prolific participants), increasing the statistical power and demonstrating the applicability to multiple populations.

5.2 Method

Participants

160 undergraduates signed up to participate in exchange for credit in an introductory psychology course. Of these, 17 voluntarily withdrew after reading the initial study requirements and consent form, and prior to assignment to condition. This study did not exclude participants on the basis of typing or multiplication test performance, resulting in a sample comparable to Study 2. However, because participants' ability to demonstrate their knowledge on our speed test was necessarily limited by their typing ability, it is worth noting that data from participants who had low typing scores (< 70% correct, 12 participants) may not be valid. (See Appendix F for analyses without those participants, although the overall findings remain unchanged). All participants who

began the study, completed it, and no participants were excluded on the basis of their answers to the follow-up questions. Upon further inspection of the data, I discovered that one participant had a single training trial repeated twice in a row (likely due to a momentary internet failure). This participant was excluded. This left a final sample of 139 (mean age = 19.6, 82 female, 2 non-binary gender).

Design

Study 4 was a single-session online experiment with participants randomly assigned to computation (n = 73) or memorization (n = 66). Study 4 was identical to Study 3 except for the following changes. First, participants were supervised by an experimenter via Zoom (as in Study 1). Second, all participants were allowed to continue on to the main study, regardless of how they performed on the typing and multiplication pre-tests (see explanation in Appendix F). Second, a different algorithm was used for the distractor problems (second training). Importantly, because I had to explain a whole new algorithm to the computation participants, including a few examples of how to apply it, the instructions preceding the second training were longer for the computation participants than for the memorization participants. This would have resulted in a longer delay between the first speed test (on the 8 # n facts) and the untimed test (retesting the 8 # n facts) in the computation group. To address this and equate the timing, the memorization participants completed a short arithmetic test (15 items) prior to the second training instructions. These problems were directly related to the example problems that the computation participants were asked to solve, but appeared in scrambled order and in pieces, so as to disguise their relationship to any algorithm (see Appendix D for a complete list of problems). This was done to ensure that not only the time but also the cognitive demands during this period were similar for both conditions. This short delay test was paced by the computer (each problem was delayed for exactly five seconds regardless of when or if the participant responded), so as to precisely control the timing and equate

it as closely as possible to the timing of the computation instructions, which were also paced by the computer.

Finally, one other change from Study 3 was that participants took a speed test on the distractor set of problems, whereas in Study 3, participants learned those problems but were never tested on them. This was done for two reasons. First, in their follow-up questions in Study 3, some participants reported being irritated that they had made the effort to learn the 7 # n problems and then weren't tested on them. Of course we never want to annoy our valuable psychology research participants. Second, this seemed like useful data to have, as it could be used to replicate Study 1's finding that the memorization advantage persisted even after learning multiple sets of problems. Essentially, the thinking was: if I'm going to have participants spend 5-10 minutes learning these problems anyway, why not have them spend an additional 2 minutes taking a test on them, so that I can analyze that data. Of note, these two changes (the different algorithm necessitating new instructions, and the test on the distractor problems) necessarily increased the delay between the first speed test and the untimed test, as compared to Study 3. It was expected that this would enhance the condition difference on the untimed test, as even more forgetting was expected to occur between the first speed test and the untimed test, although that was not the primary goal of these changes.

Materials and Apparatus

Materials and apparatus were identical to those described for Experiments 1, 2, and 3.

Procedure

The procedure was identical to Study 3 except for the changes necessarily implied by the changes to the design described above.

5.3 Results

All data were analyzed using the same models described previously.

First Speed Test

Replicating Studies 1, 2, and 3, participants in the memorization condition performed better on the first speed test than participants in the computation condition (50.4% correct versus 42.9% correct; PPE: Median = 0.56, MAD = 0.32, $P(\beta_{LC} \leq 0) = 0.05$; Figures 5.1 and 5.2).

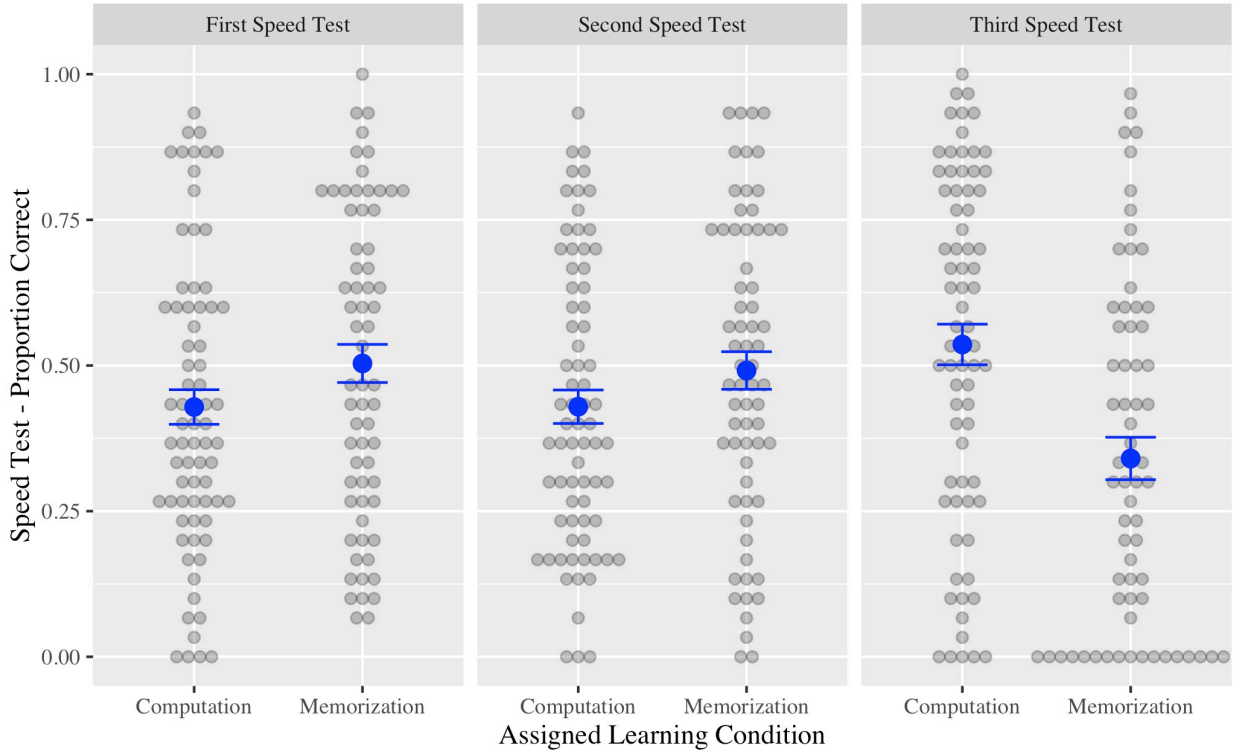

Figure 5.1. Performance on the first, second, and third speed tests.
The first, second, and third speed test tested 8 # n problems, 7~n problems, 8 # n problems again, respectively. Results are separated by assigned learning condition. For transparency and ease of interpretation, blue dots and error bars represent simple averages and standard errors of the displayed data points (i.e., each participant's proportion correct). More precise estimates of the effect are provided by the Bayesian multi-level logistic regression model with control variables (see Figure 5.2), although the overall pattern of results is the same.

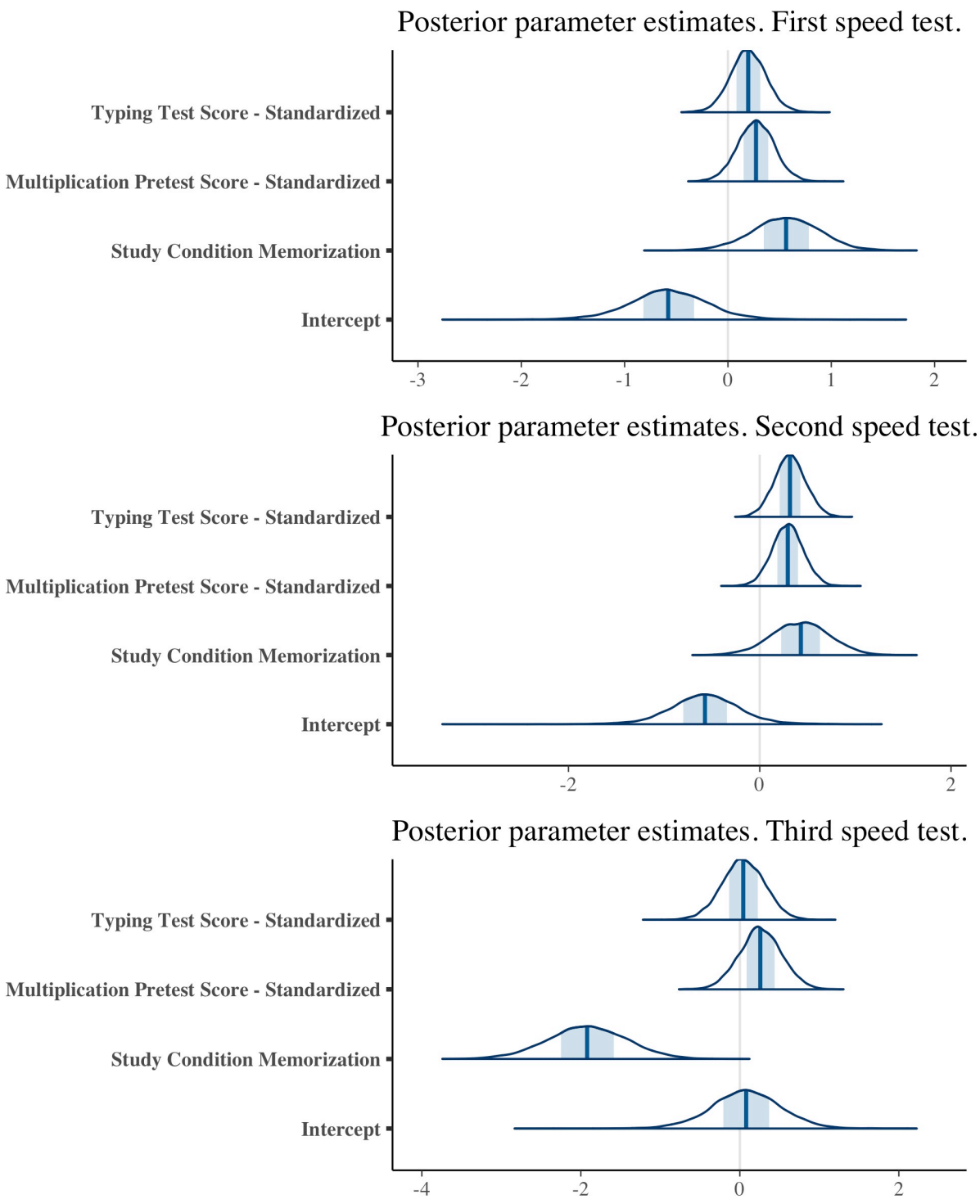

Figure 5.2. Posterior parameter estimates for the model fit to the first, second, and third speed test data.
Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Study Condition Memorization", representing how much better the memorization condition performs than the computation condition.

Second Speed Test

On the second speed test, testing the distractor 7 ~ n facts, memorization participants (49.1% correct) again outperformed computation participants (42.9% correct) in my sample. My model suggests that it is most likely that memorization does actually result in superior recall on this task, although there is a small but real chance (~8%) that computation leads to recall that is as good as or better than memorization (PPE: Median = 0.43, MAD = 0.30, $P(\beta_{LC} \leq 0 = 0.08)$; Figures 6.1 and 6.2).

First versus Second Speed Test

Although the gap between computation and memorization participants was numerically smaller on the second speed test (6.2 percentage points difference versus 7.5 percentage points difference on the first speed test), there is no clear statistical evidence that the gap between conditions shrank (PPE: Median = -0.11, MAD = 0.27, $P(\beta_{LC\times TR} \geq 0) = 0.35$). Again, as in Study 1, this should be interpreted as a lack of information about whether or not such an effect exists rather than as clear confirmation that it does not.

Third Speed Test

On the third speed test, retesting the 8 # n facts following the feedback-less practice, computation participants outperformed memorization participants (53.6 % correct versus 34.0% correct; PPE: Median = -1.92, MAD = 0.49, $P(\beta_{LC} \geq 0) = 0.0001$, Figures 5.1 and 5.2).

First versus Third Speed Test

Computation participants were able to correctly answer more items on the third speed test than the first (average change = +10.7 percentage points; PPE: median = 0.74, mad = 0.25, $P(\beta_{LC} \leq 0) = 0.006$; posterior distributions not shown), while memorization participants answered fewer items correctly (average change = - 16.3 percentage points; PPE: median = -2.13, mad = 0.52, $P(\beta_{LC} \geq 0) = 0.0006$; posterior distributions not shown). The average slope (i.e., the change from

speed test 1 to speed test 3) in the memorization condition was reliably more negative than the average slope computation condition (PPE: median = -2.41, mad = 0.37, $P(\beta_{LC\times Task} \geq 0) < 0.0001$; Figures 5.3 and 5.4)

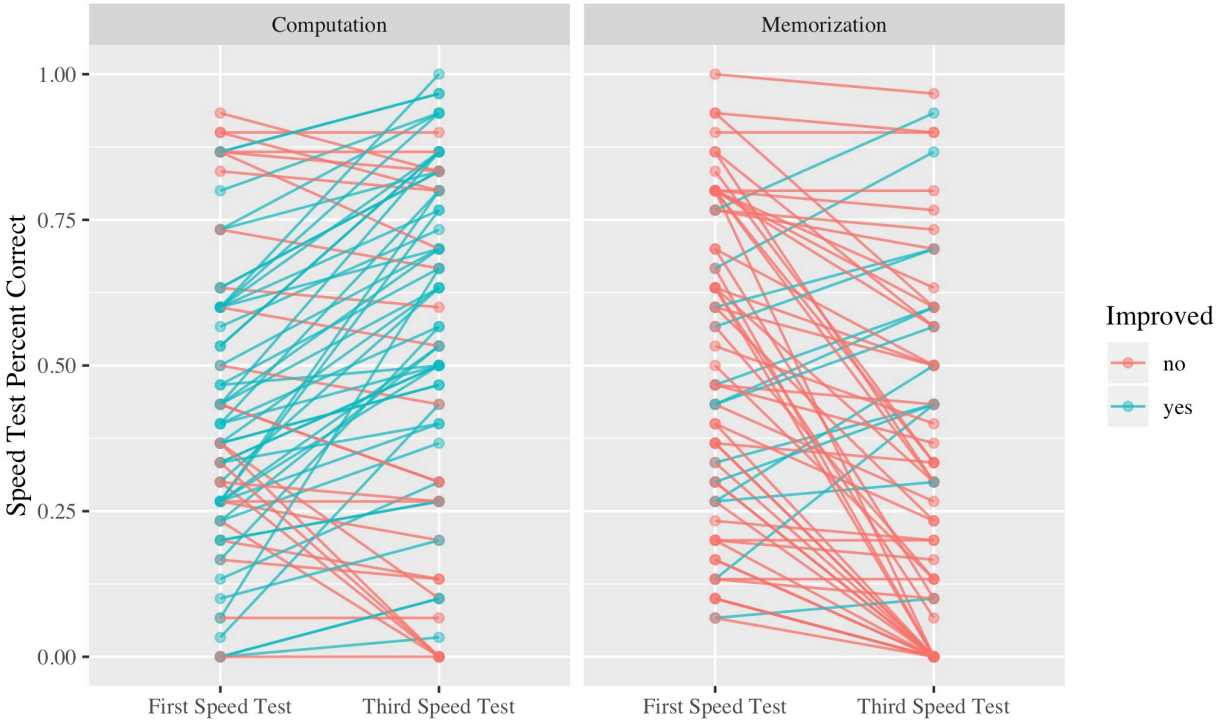

Figure 5.3. Change in performance from the first to the third speed test.
Lines connect the two test scores belonging to the same participant. Teal colored lines are used for participants who improved from the first to the third speed test, red lines for those who did not improve.

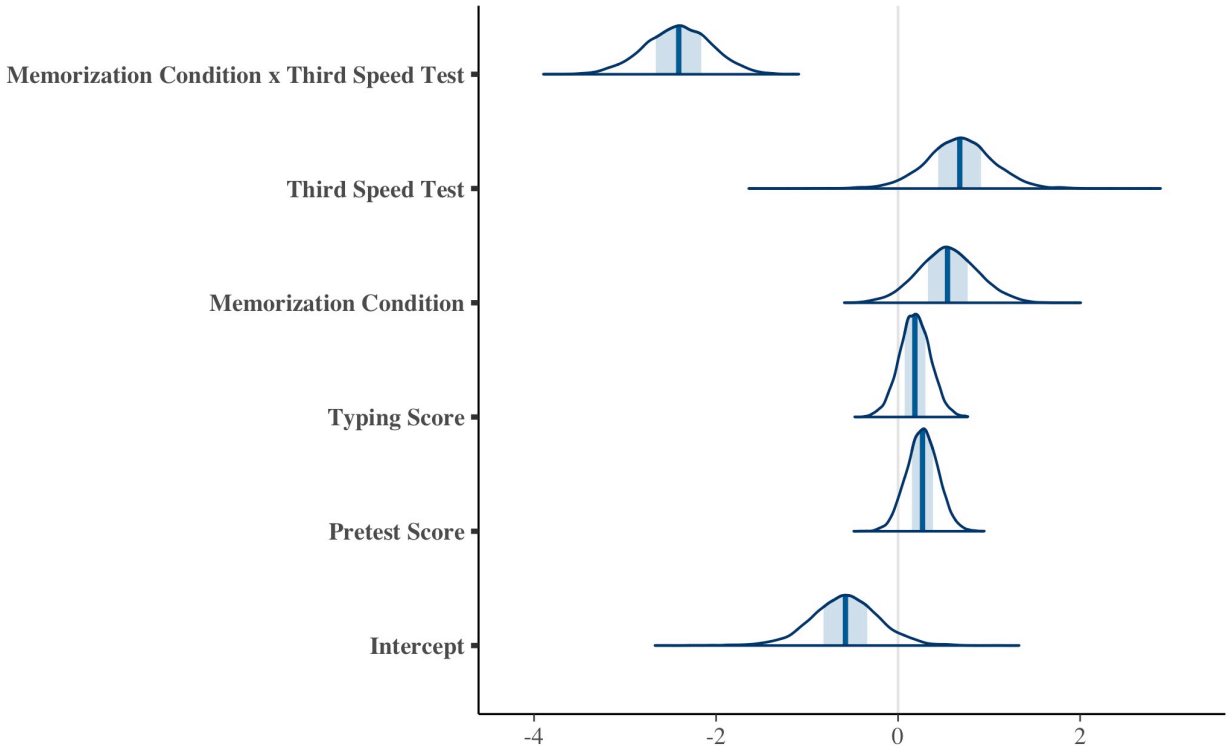

Figure 5.4. Posterior parameter estimates for the model comparing the first and third speed tests. Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Memorization Condition x Third Speed Test", representing the difference in the slope (i.e., change from test 1 to test 3) in the memorization condition versus the computation condition.

Untimed Test

Memorization participants answered significantly fewer items correctly on the untimed test as compared to computation participants (43.6% correct versus 83.2% correct; PPE: Median = -4.41, MAD = 0.70, $P(\beta_{LC} \geq 0) < 0.0001$; Figures 5.5 and 5.6).

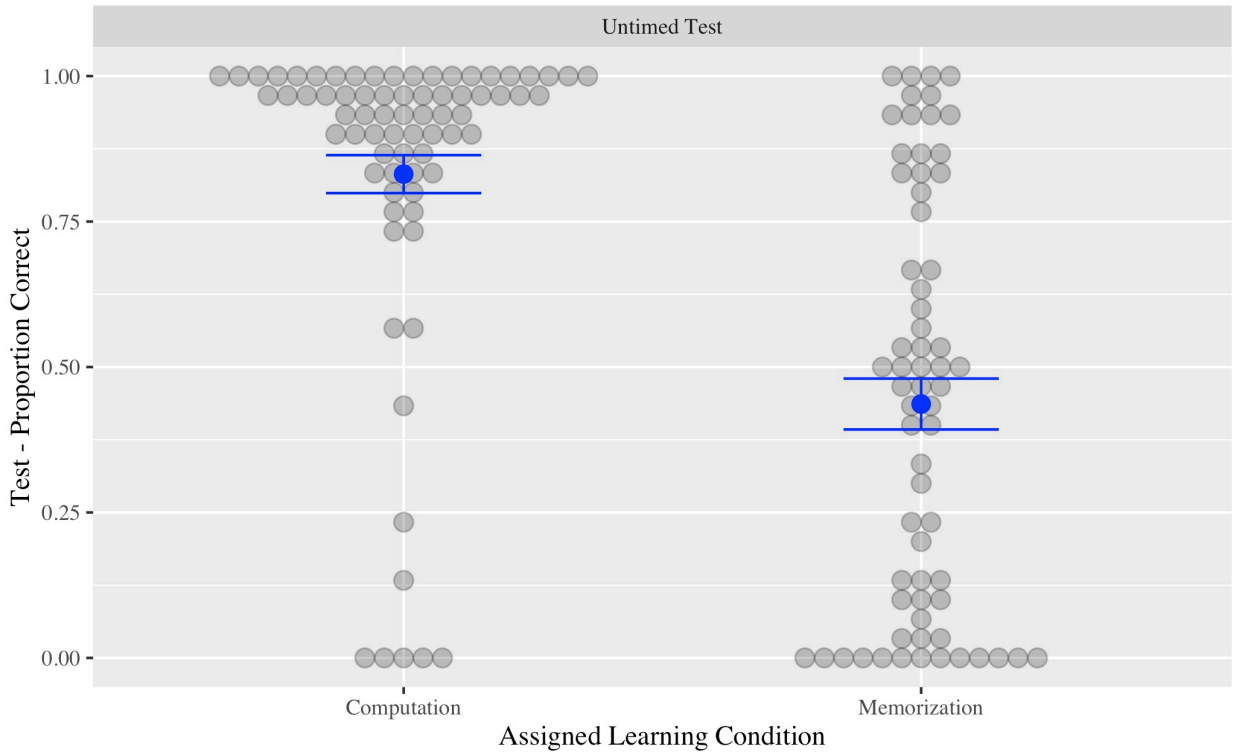

Figure 5.5. Performance on the untimed test by condition.
For transparency and ease of interpretation, blue dots and error bars represent simple averages and standard errors of the displayed data points (i.e., each participant's proportion correct). More precise estimates of the effect are provided by the Bayesian multi-level logistic regression model with control variables (see Figure 5.6), although the overall pattern of results is the same.

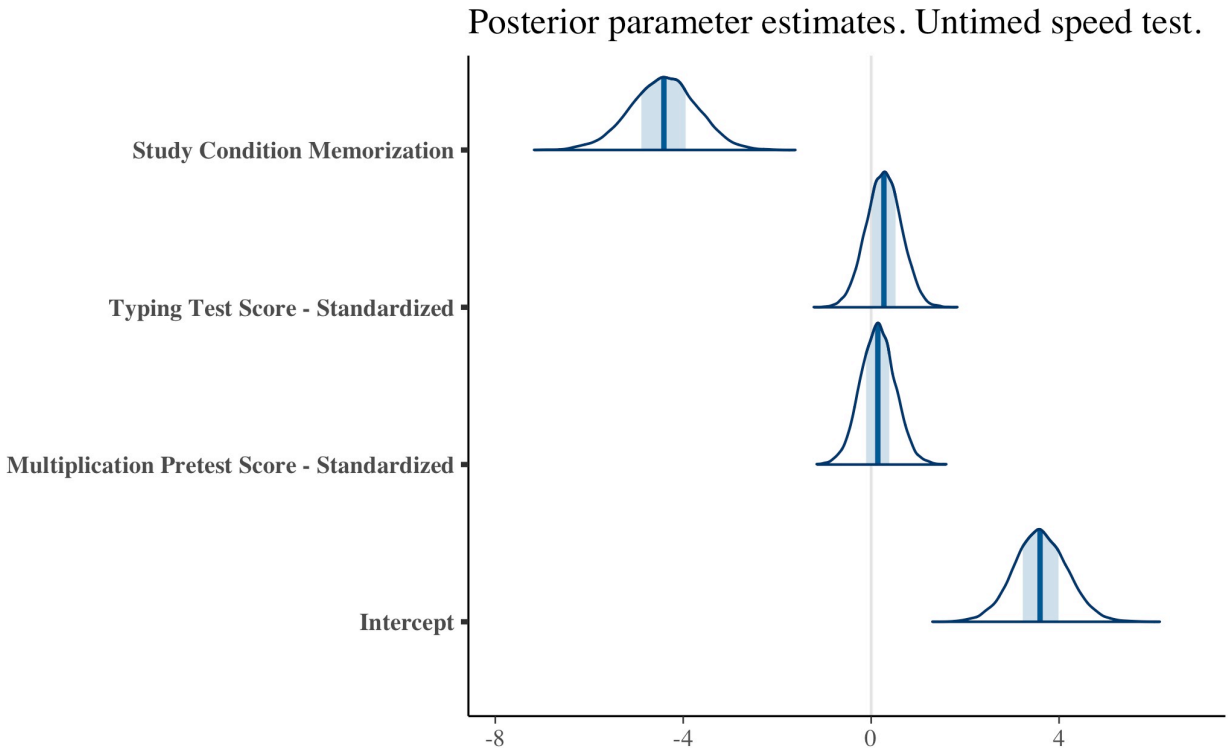

Figure 5.6. Posterior parameter estimates for the model fit to the untimed test data. Dark blue vertical lines represent medians. Light blue shaded areas represent fifty percent probability. The primary parameter of interest is "Study Condition Memorization" representing the difference in performance between the memorization and computation conditions.

Relationship Between Untimed Test Performance and Final Test Performance

Since this study has a larger sample size than Studies 3 or 4, I conducted additional analyses looking within the computation condition. I found that how well a computation participant did on the untimed test predicted how well they did on the third speed test. Specifically, as shown in Figures 5.7 and 5.8, among the eight computation participants who answered fewer than 50% of items correctly on the untimed test, not a single one improved from the first to the third speed test. By contrast, among the 65 computation participants who answered more than 50% of untimed test items correctly, 48 (74%) improved from the first to the third speed test. Of note, all eight of the low untimed test performers (less than 50% correct) also performed poorly on the first speed test

(less than 50% correct), so a fairer comparison might be to compare them to their peers who also performed poorly on the first speed test. In the subset of computation participants who performed below 50% on the first speed test, the comparison still holds, and, if anything, is even stronger; out of 38 participants who performed below 50% on the first speed test but above 50% on the untimed test, 32 of them (84%) improved from the first to the third speed test. This binning (below versus above 50%, and improved or did not improve) makes it easy to see the pattern, but the relationship also holds when analyzed continuously, i.e., among computation participants, higher untimed test performance predicts greater improvement from test 1 to test 3 (PPE: Median = 1.64, MAD = 0.27, $P(B_{TR\times UT<0}) < 0.0001$; posterior distributions not shown). The following model was used in the foregoing analysis:

$$logit(p_i) = \alpha_0 + \beta_{TT} \cdot TypingScore_i + \beta_{PT} \cdot PretestScore_i + \beta_{TR} \cdot TestRound_i + \beta_{UT} \cdot UntimedTestScore_i + \beta_{UT\times TR} \cdot UntimedTestScore_i \cdot TestRound_i + \alpha_{j[i]} + \alpha_{k[i]} + \alpha_{j[i],k[i]} + \beta_{j[i]} \cdot TestRound_i + \beta_{k[i]} \cdot TestRound_i + \beta_{j[i],k[i]} \cdot TestRound_i$$

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{\alpha_j}^2 & \rho_{\alpha_j\beta_j}\sigma_{\alpha_j}\sigma_{\beta_j} \\ \rho_{\alpha_j\beta_j}\sigma_{\alpha_j}\sigma_{\beta_j} & \sigma_{\beta_j}^2 \end{pmatrix} \right)$$

$$\begin{pmatrix} \alpha_k \\ \beta_k \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{\alpha_k}^2 & \rho_{\alpha_k\beta_k}\sigma_{\alpha_k}\sigma_{\beta_k} \\ \rho_{\alpha_k\beta_k}\sigma_{\alpha_k}\sigma_{\beta_k} & \sigma_{\beta_k}^2 \end{pmatrix} \right)$$

$$\begin{pmatrix} \alpha_{j,k} \\ \beta_{j,k} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_{\alpha_{j,k}}^2 & \rho_{\alpha_{j,k}\beta_{j,k}}\sigma_{\alpha_{j,k}}\sigma_{\beta_{j,k}} \\ \rho_{\alpha_{j,k}\beta_{j,k}}\sigma_{\alpha_{j,k}}\sigma_{\beta_{j,k}} & \sigma_{\beta_{j,k}}^2 \end{pmatrix} \right)$$

(Observations, *i*, are nested within Participants, *j*, and Problems, *k*)

In this model $UntimedTestScore_i$ was calculated as percent correct for each participant and then standardized (i.e., z-scored) across participants (as was done in all models for the typing and pretest scores). This model was fit to the data from the first and third test rounds only.

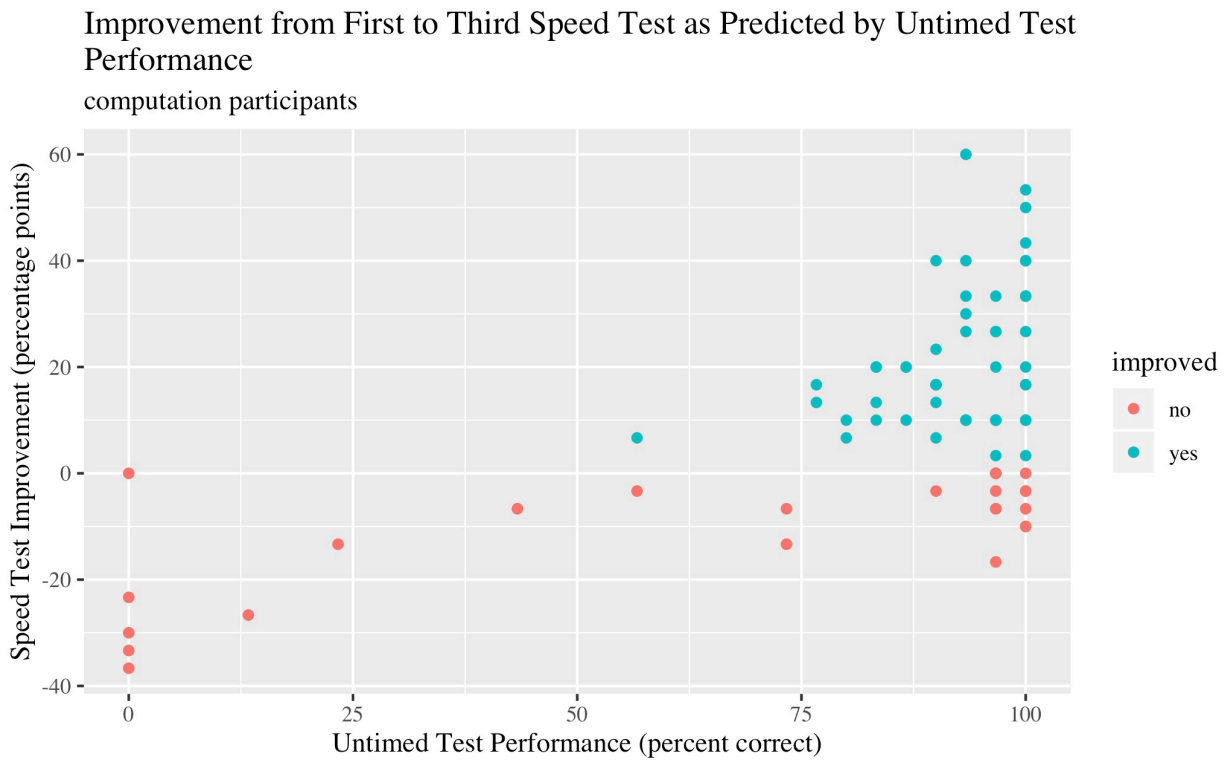

Figure 5.7 Improvement from the first to the third speed test as predicted by untimed test performance.
Each dot represents an individual computation participant. The x-axis shows how well that participant performed on the untimed test. The y-axis shows how much their score changed from the first to the third speed test. Red dots indicate participants who did not improve from the first to the third speed test (i.e., percentage point change less than or equal to zero), while blue dots indicate those who did improve. Plotting "improvement" is very useful but obscures how well participants did in the first place (e.g., someone who declined zero percentage points from an original score of 0% correct appears the same as someone who declined zero percentage points from an original score of 90% correct). To reveal that aspect of the data, Figure 5.8 plots the same data in a different way.

Figure 5.8. Improvement from the first to the third speed test as predicted by untimed test performance (binned).
This figure plots the same data as Figure 5.7, except that now "untimed test performance" is loosely binned into "below 50%" and "above 50%" rather than being displayed on the x-axis. This frees the x-axis to display a participant's actual score on each of the two speed-tests. "Improvement" (on the y-axis in figure 5.7) is now revealed by the slope of the line connecting a participant's two scores. As in Figure 5.7, blue indicates improvement, while red indicates no improvement. Since the "below 50%" group all scored below 50% correct on the first speed test, the most relevant comparison is to participants in the "above 50%" group who also scored below 50% correct on the first speed test; points and lines from participants above this threshold are faded.

Follow-Up Questions

At the end of each of my four studies, participants answered a series of follow-up questions that asked them about any patterns that they noticed, any strategies that they used, and anything else interesting they noticed about how they learned (see Appendix E for the full list of follow-up questions). In this study, only, I also choose to analyze the answers to these follow-up questions. I did not do this for Studies 1-3 because they all had smaller sample sizes; here I had 139 participants

while, in Studies 1, 2, and 3 the sample sizes were 86, 78, and 61, respectively. Since the follow-up questions were open-ended, leading to a diversity of responses, I worried that with smaller sample sizes, further divided into two conditions, and into a variety of response categories, I might end up with two few responses in any particular cell to do any meaningful analysis.

I was particularly interested in the strategies that participants reported using, where "strategies" were anything that participants reported intentionally doing in order to more successfully answer the questions during the practice or test rounds. To classify participants' strategies, I first blinded myself to participants' assigned conditions. I then read through participants' answers to the questions and made a list of all different classes of strategies that appeared. Any response that was given by five or more participants (regardless of condition) was selected as an official response category, and I then read through all of the responses again (still blinded) and assigned each participant's responses to these categories. Of note, since some participants mentioned multiple strategies, a single participant might be counted in multiple response categories.

The strategies reported are displayed in Figure 5.9 and the reader is encourage to reference that figure while reading this section as, for the sake of brevity, I do not list every number individually in the text.

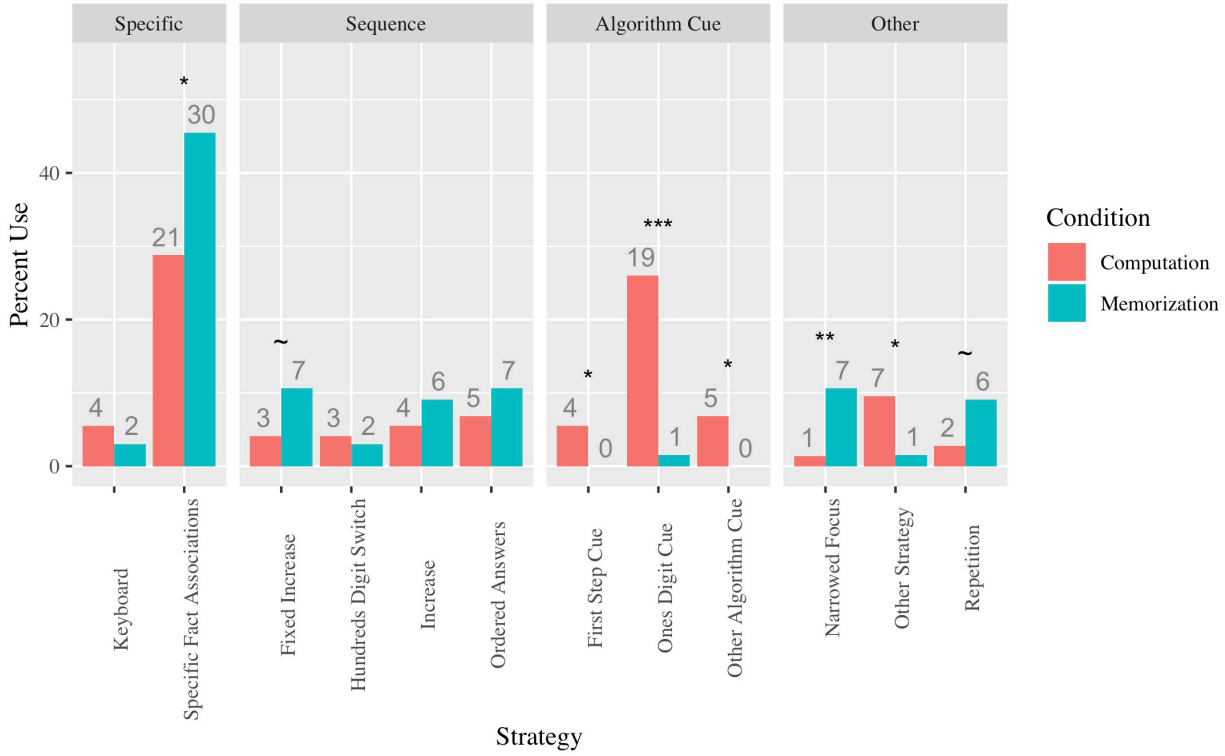

Figure 5.9. Reported strategy use by condition.
Similar strategies were binned into larger groups (panels). The numbers above each bar are raw counts, e.g., the first "4" on the left indicates that four computation participants reported using a "keyboard" strategy. The height of the bars is based on percent use in each condition, and as there were not the same number of participants in both conditions (memorization = 66, computation = 73), the height of the bars may not be directly proportion to the counts at the top of each bar. Analyses were Bayesian, so p-values do not have the standard interpretation, but asterisks are used to give a sense of the magnitude of the Bayesian p-values (*** = $p < 0.001$, ** = $p < 0.01$, * = $p < 0.05$, ~ = $p < 0.10$).

The most commonly reported strategy in both conditions was what I termed "specific fact associations." This referred to the strategy of looking for meaningful associations that could be used to remember a single answer or to relate two answers. Specific fact associations generally fell into three sub-categories (not separately listed in Figure 5.9). Some were based on the answer itself, for example noticing that "134" could be thought of as a simple addition equation (1 + 3 = 4), recognizing that "225" was a known square (15 squared), or relating a particular answer, such as "206" to a familiar area code. Other specific fact associations related the numbers in the question to

the numbers in the answer, for example noting that in 7 ~ 6 = 167, the "6" and "7" from the question appeared in the answer but in reversed order, or noticing that in 8 # 8 = 224, the digits in the answer sum to 8, which is the digit in the question. Finally, some specific fact associations related two answers, for example noticing that the answers to 8 # 8 (224) and 8 # 9 (242) contained the same digits but in different orders. Memorization participants were more likely than computation participants to report using specific fact associations ($P(\beta_{LC} \leq 0) = 0.02$). Another related kind of specific association was that some participants reported noticing a pattern in the location of the keyboard keys used to answer a specific question. Reported use of this keyboard strategy did not differ significantly by condition.

Participants also reported using several strategies based on more general fact patterns, i.e. patterns which held across *all* problems in a set. I separated these into two broad categories: "sequence" based strategies and "algorithm cue" strategies (second and third panels in Figure 5.9). Sequence based strategies involved ordering the answers from least to greatest. These were further subdivided into four specific reported strategies. As n increased by 1, the answers to the 8 # n facts increased by 18 and the answers to the 7 ~ n facts increased by 39. For example 8 # 3 was 134 and 8 # 4 was 152, a difference of 18. Participants who reported using this precise difference between successive answers to help recall the answers for one or both fact sets were counted as using a "fixed increase" strategy, and this was somewhat more commonly reported among memorization participants than computation participants ($P(\beta_{LC} \leq 0) = 0.07$). Participants who reported knowing that the answers to 8 # n or 7 ~ n increased with increasing n and using that to help narrow the range of possible answers to each problem, but who did not specify that there was a constant increase between successive problems, were coded as using an "increase" strategy. Reported use of this strategy did not differ by condition. Relatedly, some participants stated that they deliberately re-ordered the facts in their minds, specifically attempting to learn them in order from least to greatest, even though they were presented in scrambled order, an "ordered answers" strategy. Again this did

not differ by condition. Finally, another sequence-based strategy that several participants reported using was paying attention to the n-value above which all answers would be in the 200s rather than the 100s. For example, 8 # 3 through 8 # 6 had answers in the 100s, and 8 # 7 through 8 # 9 has answers in the 200s. Reported use of this strategy did not differ by condition.

The other type of general pattern strategy was "algorithm cue" strategies. These strategies all used some piece of the algorithm, or a modified version of the algorithm, as a cue to facilitate recall of the answer. The most common of these strategies involved figuring out the ones digit of the answer, and then using that to trigger recall of the full answer. For the 8 # n facts, this could be done by multiplying 8 x n and noticing that the ones digit of that product was equal to the ones digit of the answer. For example, for 8 # 3, 8 x 3 = 24, and the full answer is 154. Both have a "4" as their ones digit. This strategy can be successfully used across all 8 # n facts. Similarly, for the 7 ~ n facts, the ones digit of the answer was equal to ten minus the ones digit of 7 + n. For example, for 7 ~ 4, 7 + 4 is 11, which has a ones digit of 1, and the overall answer is 109, which has a ones digit of 10 - 1, i.e., 9. Again, this held across all 7 ~ n facts. Unsurprisingly, this strategy was reported much more frequently among the computation participants ($P(\beta_{LC} \leq 0) < 0.0001$). A few computation participants also reported using the first step of the algorithm to help facilitate recall of the answer. For example, for the 8 # n problems, the algorithm was 10(8 + n) - 8n. A participant trying to recall 8 # 3, might execute just the first 10(8 + n) step, getting 110, and then know that the answer had to be a bit more than 110. By contrast the answer to 8 # 9 had to be at least 170 (i.e., 10 x (8 + 9)), eliminating several candidate answers. Finally, few additional computation participants reported using some other part of the algorithm to help them, or generally reported using a part of the algorithm but did not specify which part. These participants are counted under the "other algorithm cue" category. No memorization participant fell into either of these last two categories.

Finally, there were some strategies that were not pattern or association based at all. These were generally grouped under a broad "other" category. One commonly reported strategy in this

category was intentionally trying to focus on learning just one or two of the facts at a time, ignoring the rest, and then focusing on additional facts once the first ones were mastered. This strategy was reported more often by memorization participants than computation participants ($P(\beta_{LC} \leq 0) = 0.01$). The second most common non-pattern-based strategy was deliberately repeating the answers (with or without the corresponding questions) in an attempt to memorize them. This was used by more memorization than computation participants ($P(\beta_{LC} \leq 0) = 0.05$). Finally, any participant who mentioned a strategy that was reported by too few participants to be counted as its own distinct strategy was lumped into an "other strategy" category. Participants whose answers clearly indicated strategy use but were difficult to interpret were also included in this category. More computation than memorization participants reported strategies that fell into this category $P(\beta_{LC} \leq 0) = 0.01$). Some example responses from this category were, (a) "I would "write" in the air the word next to the computation and have that glued to my mind," (b) "i used acronyms to help memorize," (c) "I have color grapheme synesthesia, so… I was able to keep a color combination present for my answer," and (d) "I tried to memorize the numbers as bits of like a password."

Finally, I examined whether reported use of certain strategies correlated with performance on either the first or the final speed test. For example, I asked whether computation participants who reported using "specific fact associations" performed better on the tests than computation participants who did not report using such a strategy. Analyses with very few participants in a given category are likely to be underpowered, so I limited myself to only looking at those cases where at least 5 participants in a condition reported using a given strategy. This meant that I analyzed two strategy categories in the computation condition: "specific fact associations" and "ones digit cue", and six categories in the memorization condition: "specific fact associations", "fixed increase," "increase," "ordered answers," "narrowed focus," and "repetition." All of these analyses were conducted using the following model:

$$logit(p_i) = \alpha_0 + \beta_{TT} \cdot TypingScore_i + \beta_{PT} \cdot PretestScore_i + \beta_S \cdot UsedStrategy_i + \alpha_{j[i]} + \alpha_{k[i]} + \alpha_{j[i],k[i]}$$

$$\alpha_j \sim Normal\left(0, \sigma_j^2\right)$$

$$\alpha_k \sim Normal\left(0, \sigma_k^2\right)$$

$$\alpha_{j,k} \sim Normal\left(0, \sigma_{j,k}^2\right)$$

(Observations, *i*, are nested within Participants, *j*, and Problems, *k*)

This model was fit separately to the data from each of the three speed tests, and was also run separately for each strategy category of interest, each time with the $UsedStrategy_i$ variable indicating whether or not the participant reported using the strategy of interest. As there are a large number of tests here, and these analyses were not particularly relevant to my central research questions, rather than talk through each of these results, I simply refer the reader to table 5.1. Overall, the major findings were as follows. Among computation participants, reported use of a "specific facts association" strategy was correlated with better performance on all three speed tests and use of a "ones digit cue" strategy was correlated with better performance on the third speed test, and somewhat correlated with better performance on the other two speed tests. . Among memorization participants, reported used of an "increase" strategy was correlated with better speed test performance. No other memorization strategy was reliably correlated with speed test performance.

			Test 1			Test 2			Test 3		
		Count	Median	MAD	$P(B_S \leq 0)$	Median	MAD	$P(B_S \leq 0)$	Median	MAD	$P(B_S \leq 0)$
COMPUTATION	**Specific Fact Associations**	21	1.06	0.50	**0.02**	0.94	0.45	**0.02**	1.37	0.58	**0.009**
	Ones Digit Cue	19	0.79	0.53	*0.07*	0.65	0.48	*0.09*	1.69	0.58	**0.002**
MEMORIZATION	**Specific Fact Associations**	30	-0.07	0.45	0.57	0.38	0.43	0.18	1.49	0.96	*0.06*
	Fixed Increase	7	0.67	0.69	0.17	0.98	0.65	*0.07*	0.20	1.39	0.44
	Increase	6	1.96	0.73	**0.004**	1.81	0.67	**0.006**	2.83	1.33	**0.02**
	Ordered Answers	7	0.01	0.69	0.49	-0.03	0.65	0.52	0.29	1.37	0.41
	Narrowed Focus	7	-0.99	0.70	*0.92*	-0.93	0.69	*0.91*	-0.90	1.39	0.74
	Repetition	6	-0.19	0.73	0.60	-0.03	0.70	0.52	-0.08	1.42	0.52

Table 5.1 Results from the Models Predicting Test Performance by Strategy Use
Results of the models predicting test performance by strategy use. Results are reported for the β_S parameter, only, which can be interpreted as how much better participants who used the given strategy performed as compared to other participants in the same condition who did not use the strategy. $P(\beta_S \leq 0)$ is a Bayesian "p-value" and has a different interpretation than a standard p-value (see Appendix A), but to give some sense of which results suggest a reliable relationship, values are bolded when $P(\beta_S \leq 0)$ is below 0.05 or above 0.95, and italicized when they are below 0.10 or above 0.90.

5.4 Discussion

In this study, participants learned a set of facts, took a speed test on it, learned a second set of facts, took a speed test on those facts, completed an untimed-test on the first set of facts, and then took a third speed test, retesting the first set of facts. Once again, memorization participants outperformed computation participants on the first speed test (50% versus 43% correct), replicating the effect reported in all three previous studies. Furthermore, despite initially recalling more of the

facts, memorization participants performed worse on the untimed test (44% versus 83% correct for the computation participants), which retested their ability to answer the first set of facts after a delay and without time pressure.

Why did memorization participants perform worse on the untimed test? The most likely explanation is that participants in both conditions forgot many of the answers by the time they reached the untimed test. For problems whose answers were forgotten, computation participants could recompute the forgotten answers, while memorization participants had no recourse. This result is not as trivial as it first seems. In order to recompute the answer, computation participants had to accurately remember the multistep algorithm (a # b = 10(a + b) + ab). This was particularly difficult in this study, because participants had just practiced a different, potentially interfering algorithm (a ~ b = 30b - (a + b)). (The use of this second interfering algorithm is what sets this study apart from Study 3). Indeed, I was surprised by how many of the computation participants correctly remembered the first algorithm. I fully expected that my results on the untimed test might be bimodal with a significant subset of computation participants completely forgetting the algorithm and answering very few items correctly. In actuality, the vast majority of the computation participants appear to have remembered the algorithm (see Figure 5.5).

Digging deeper, we might ask why computation participants were largely successful at remembering the algorithm when memorization participants had difficulty remembering the six individual answers over the same delay (or actually, a shorter delay for most memorization participants, as they tended to complete the second training faster than computation participants). One simple, and powerful explanation is that the computation participants had fewer pieces of information to remember. While they needed to recall only a single algorithm (or, perhaps more accurately, a set of three steps), the memorization participants needed to recall six distinct answers (or, as each was three digits, eighteen digits).

Of course the number of pieces of information may not be the only reason why the computation participants were so successful at remembering the algorithm. A second obvious candidate explanation is that the algorithm was more frequently practiced than each of the answers. That is, in the computation condition, the algorithm was used on *every* training problem (or at least initially - later some answers may have been recalled without computation) for a total of up to 48 trials, while, in the memorization condition, each individual answer appeared on only one sixth of the problems, or 8 trials.

I was also surprised to observe some additional behavior that might account for why computation participants were so successful on the untimed test. Anecdotally, I noted that some computation participants who spontaneously thought aloud as they completed the task appeared to be using the few answers that they did recall to assist them in pinning down the correct algorithm. For example, a participant who was uncertain if perhaps the algorithm was a # b = 30a + ab, might be absolutely certain that they remembered 8 # 7 = 206. A quick computation would reveal that under their proposed algorithm, 8 # 7 = 296. Since the participant knew that 8 # 7 = 206, they would reject that potential algorithm and try another algorithm until they found one that worked. That is, they not only used the algorithm (if they recalled it) to re-derive the correct answers, as I anticipated, but they also used any answers that they recalled to help re-derive the algorithm.

My next finding was that computation participants outperformed memorization participants on the third speed test, which retested the 8 # n problems after the untimed test (54% versus 34% correct). Again, this replicated Study 3, but the result here was even stronger and more reliable, likely due to both the larger sample size and the longer delay prior to the untimed test in this study. Once again, I propose that this superior performance on the final speed test is a direct result of participants' experience on the untimed test. That is, computation participants were able to recompute any forgotten answers on the untimed test, and this re-exposure to the correct answers

boosted their subsequent memory for those answers, resulting in superior performance on the final speed test.

Supporting this proposal, I found that computation participants who performed better on the untimed test showed a greater improvement from the first to the final speed test, and that computation participants who performed especially poorly on the untimed test (below 50%), indicating that they either could not recall or could not effectively apply the algorithm, showed no improvement at all. It should be noted, however, that this is correlational data, i.e., participants weren't manipulated in some way to cause some of them to perform poorly on the untimed test, and like all correlational data should be interpreted with caution. It remains possible that some other underlying participant characteristic (e.g., attention, intelligence, effort) is driving both poor untimed test performance and lack of improvement on the speed tests. At a minimum, though, these results are certainly consistent with my hypothesis, and if the converse were observed, i.e., if we were to see participants who performed poorly on the untimed test but still improved from the first to the third speed test, which we do not, then that would be evidence against my proposed mechanism.

Finally, I also note that on my second speed test, testing the 7 ~ n facts, memorization participants outperformed computation participants (49% versus 43% correct) in line with all of my other results, however this difference was not as large or as reliable as some of the others I have seen (according to my posterior parameter estimates, there is a 8% chance that computation participants actually perform as well as or better than memorization participants). This is consistent with my findings from Study 1 where the gap between computation and memorization was also slightly smaller and less reliable on the second speed test, testing a different set of facts. Although there are a number of logical explanations for why it would be the case that the gap between the two conditions would shrink from the first to the second speed test, I will avoid the temptation to speculate about them here, as there is no clear statistical evidence that the gap did in fact shrink.

Unlike the previous studies, in this study, I also chose to analyze participants' open ended responses to the follow-up questions that appeared after the final speed test. In particular, I focused on the strategies that participants reported using. One caveat that should be kept in mind when interpreting these results is that I measured *reported* strategy used, not actual strategy use. To the extent that participants in a given condition were more likely to think that they were *supposed* to (or in some cases explicitly *not* supposed to) use a particular strategy they may be more (or less) likely to report using it. That is, they may be telling me what they think I want to hear rather than what they actually did. Furthermore, It should be noted that my tallies likely represent undercounts of actual strategy use, as the open ended nature of the questions meant that participants may not have written down every strategy that they used, feeling that some were too trivial to report, or not even thinking of some of the things that they did as qualifying as "strategies" per se. Additionally, some participants who used multiple strategies may not have been motivated enough to take the time to list them all. That said, it should still be the case that the strategies that were the most frequently used and/or the most salient to the participants should have been reported most often.

One result of these strategy analyses that is not novel but bears keeping in mind, is that memorization participants reported engaging in a fair bit of deliberate elaboration as they attempted to learn the material. That is, when we think of pure "rote memorization", we might imagine that participants are simply starting at the answers or repeating them over and over. While some memorization participants did report repeating the answers (9%), many more reported intentionally elaborating or reorganizing the given facts to make them more meaningful. For example, memorization participants reported deliberately learning the answers in order from smallest to largest even though they were presented in scrambled order (11%), making use of the fact that the answers increased by a fixed amount each time the second operand increased by one (11%), and noticing unique features of individual answers that made them more memorable (41%) such as that there was a "3" in the answer to 8 # 3, that the answers to 8 # 9 and 9 # 8 contained the same three

digits in different orders, or that an answer was related to a number that was personally meaningful to them, among other useful associations.

Similarly, the computation participants did more than just execute the computation and input the answer. Even though they were not directed to memorize the answers, a number of computation participants reported using strategies that were clearly aimed at deliberate memorization such as ordering the answers from smallest to largest (7%), repeating the answers to themselves (3%), focusing on learning just one or a few problems first (1%), or noticing and applying the same sorts of useful associations (29%) that were described above for the memorization participants (like the "3" in the answer to 8 # 3). This more detailed look at what participants actually experienced in each of the conditions, which may be quite different from what we might have imagined, can help constrain the possible mechanisms that might actually be driving the observed differences in final performance.

Another noteworthy result was that a significant number of computation participants reported noticing that the final digit of the answer could be rapidly computed from the two given operands, and using that final digit as a cue to recall the entire answer. Interestingly, while a few memorization participants noticed (6%) and made use of (2%) the fact that the final digit could be computed using the tricks above, this strategy was far less common than it was in the computation condition. This suggests that one benefit of computation may be that it makes useful patterns in the problems more salient. That is, if I am aware that a # b = 10(a + b) + ab, then I am more likely to notice that logically it must be the case that the ones digit of the answer is the same as the ones digit of ab. Such a relationship should be significantly less obvious for a memorization participant, who would have to happen to notice it by chance. Similarly, an elementary student who has learned to compute 6 x n as 5 x n + n (e.g., 6 x 8 = 5 x 8 + 8), may be more likely to notice that when multiplying 6 by any single digit even number, the answer always has half that number in the tens place, and that number itself in the ones place, i.e., 6 x 2 = 12, 6 x 4 = 24, 6 x 6 = 36, 6 x 8 = 48).

This logically follows from the fact that, when n is even, 5 x n is always half of n in the tens place and zero in the ones place, and then 6 x n is just 5 x n + n. Or, as an even simpler example, a student who simply learns additional facts by rote memorization may happen to notice that, for single digit numbers, the answer to 8 + n has a 1 in the tens place and has 2 less than n in the ones place. For example, you can find the answer to 8 + 7 by writing a 1 and then following it with the number that is two less than 7, for a final answer of 15. However, this pattern may be more obvious to a student who intentionally learns to compute 8 + n facts using the algorithm 8 + n = 10 + n - 2, Overall, then, I propose that computation may make other meaningful patterns in the answers more salient. To the extent that recognizing these patterns makes it easier to recall the correct answers, this could represent an additional mechanism via which computation benefits memory.

On the other hand, somewhat more memorization participants reported noticing and using the fixed increase between successive problems (11% memorization versus 4% computation), i.e., that each time n increased by 1, the answer to 8 # n increased by 18 and the answer to 7 ~ n increased by 39. One reasonable conclusion from these results might be that the salience of patterns that derive directly from the algorithm itself (like the final digit tricks) may prevent participants from noticing other useful patterns (like the fixed increase). It's not clear, however, that that is to the computation participants' detriment overall. The total number of computation participants who used either algorithm-based patterns or the increase based patterns was still greater than the combined total of memorization participants who noticed either of these patterns. That is, even though computation participants may be less likely to notice non-algorithm based patterns, they may be much more likely to notice algorithm-based patterns, leading to more useful pattern recognition overall.

Finally, I also analyzed whether participants who reported using certain strategies were more successful at learning the problems. We should be very careful about reading too much into these results, as this is a correlational sub-analysis embedded in an otherwise experimental study. That is, it

is important to remember that I did not deliberately manipulate strategy use. As such, it is entirely possible that additional confounding variable(s) drive any observed relationship between strategy use and test performance. For example, it is possible that participants who exert more effort or who are more intelligent, may be both more likely to report using certain clever strategies and independently more likely to remember many facts. A separate, but equally important issue, is that it is possible that certain strategies are particularly well suited to certain learners. That is, each learner may naturally select the strategies that work best for them, and it may be the case that if another learner were directed to use that same strategy, it would not improve their memory. For these reasons, I would caution strongly against concluding that if a particular strategy was correlated with better performance in this study, then we should encourage all learners to use that strategy. That said, the strategies that *are* correlated with better performance here are at least good candidates for follow-up studies that expressly manipulate strategy use.

Overall, I found that, on all three speed tests, computation participants who reported using "specific fact associations" performed better than computation participants who did not (though this effect is less strong on the second speed test). Additionally, computation participants who used a "ones digit cue" tended to perform better on the speed tests, and this was very much the case for the third speed test in particular. Among memorization participants, the specific fact associations strategy only predicted better success on the third speed test, and even there, not particularly strongly. Although I hesitate to make too much of any result that has not been consistently replicated and that is purely correlational, I will note that logically, at least, this finding makes a lot of sense. It is likely that on earlier speed tests, direct recall memory for the answers was strong due to having just engaged in extensive retrieval practice with those answers. On the untimed test, when direct recall memory for the answers may have faded, participants would have had to rely more on any additional associations that they generated in order to recreate the forgotten facts, which in turn, should have affected performance on the third speed test. It is not surprising, therefore, that a

specific fact associations strategy would primarily affect third speed test performance. Participants who noticed a "fixed increase" between successive facts performed better on the second speed test only, and, again, the evidence that they did so is a bit weak. The strategy that most strongly predicted test performance among the memorization participants was "increase," with participants who reported using this strategy consistently performing better on all three tests. Two other strategies: "ordered answers" and "repetition" seemed to have no discernible relationship to test performance among memorization participants. Finally, participants who reported using a "narrowed focus" strategy trended towards actually performing worse on the speed tests than participants who did not. Although, again, since these data are correlational, it is difficult to know whether narrowing one's focus to learn one or two facts at a time is actually a bad strategy that causes a participant to learn less or whether participants who particularly struggle with memorization in the first place were more likely to adopt this strategy as a way of making the task less overwhelming.

CHAPTER SIX: General Discussion

This series of studies asked whether learning arithmetic facts via self-computation or rote memorization would result in better arithmetic fact fluency. Prior attempts to test this have produced mixed results, with most lab-based studies finding no difference or an advantage of memorization and most classroom studies reporting an advantage of computation. The present set of studies clarifies why this effect has been difficult to pin down, helps predict the circumstances under which each study method is expected to be successful, and also provides a more general framework for thinking about teaching and learning.

6.1 Flashcard Like Memorization Results in Better Immediate Recall Memory than Self-Computation

Across four studies, I consistently found that immediately after studying six artificial arithmetic facts, flashcard-like memorization led to better ability to rapidly recall the studied facts than did self-computation. Importantly, although the four experiments were similar, they were not identical, further strengthening the result, which held across several different participant samples, different problem sets, different algorithms, different amounts of practice, and with and without experimenter supervision. Experiments 1 and 4 further demonstrated that this memorization advantage persisted (although somewhat attenuated) when a second set of six facts was learned. It is also worth noting that memorization was not only more effective, but also more efficient in all experiments, with participants not only learning more facts, but taking less time to do so. Furthermore, I note that the computation condition used in these experiments is likely more effective than typical classroom computation practice. I required participants to recompute an answer immediately when incorrect, while in typical paper and pencil-based practice, errors may go uncorrected, further decreasing the efficacy of computation practice relative to memorization practice in real world settings, and making the effect I found even stronger.

Overall these results support the conclusion that people *do* remember some of the answers that they compute, even if they are not deliberately trying to remember them. This aligns with research in developmental psychology, which suggests that with repeated practice computing answers (e.g., practicing computing 5 + 3 by counting on "6 ,7 ,8"), students naturally commit those answers to memory, without specific focused effort directed at memorization (e.g., Carpenter and Moser, 1988), and with research in memory and learning that suggests that, with practice, learners naturally shift from algorithmic computation to direct recall (e.g., Logan and Klapp, 1991). However, importantly, it should also be noted that while computation participants *do* develop the ability to recall some answers as a result of repeatedly computing them, they are not able to recall as many of them as their peers who engaged in flashcard-like practice. That is, computation practice *can* lead to memory for the individual answers, but it is not as efficient as flashcard-based studying, or, at least, this is the case when participants are tested immediately after study.

As noted in Chapter 1, the two conditions used in this set of studies differ on too many dimensions to make clear inferences about *why* this difference exists. That is, we cannot say exactly why memorization leads to better immediate recall. However, my results do provide useful information that can inform educational practice: in the limited case where we care only about immediate recall memory (e.g., perhaps when cramming for an exam), I would recommend that students study with flashcards over studying via self-computation. Furthermore, it is reasonable to assume that this finding applies not only to arithmetic facts, but to other classes of information as well. For example, if the goal is immediate recall, flashcard-like practice should also be better than elaborative study for learning other fact based information, e.g., for learning science or history concepts. This broader conclusion is consistent with a series of experiments by Karpicke and colleagues, who report that retrieval practice is superior to several different elaborative encoding activities, including creating concept maps (Karpicke and Blunt, 2011), creating mental images

(Karpicke & Smith 2012), and generating semantic associates (Karpicke & Smith 2012; Lehman, Smith, & Karpicke, 2014).

6.2 The Self-re-presentation Hypothesis

Importantly, Study 1 differed in an important way from real-world arithmetic fact learning. In real classrooms, students are often called upon to use their studied facts without the benefit of immediate corrective feedback. For example, when a student completes a homework assignment or a test, they may incorrectly recall a fact or fail to recall a fact. Such an error is unlikely to be immediately corrected, and may never be corrected; even if a teacher grades the work, the student may not review it when it is returned. I hypothesized that one advantage of learning via computation might be that if a computation student cannot recall the answer to a given problem, they will likely recompute it, thereby re-exposing themself to the correct answer and strengthening their subsequent ability to recall that answer. By contrast, when a memorization participant fails to recall an answer, they will be more likely to provide no answer or to guess the answer. Omitting the answer would make them no more likely to correctly respond to the problem in the future, and responding incorrectly should actively interfere with future recall (Siegler, 1988). Overall, then, I expected that self-computation practice would be self-reinforcing, with participants strengthening their own recall ability over time as they continued to compute any answers that they could not recall. By contrast participants who learned via rote memorization should be perpetually dependent on *external* reinforcement to strengthen their memory, and in the absence of that external reinforcement, their memory for the studied facts should deteriorate over time. I refer to this as the "self-re-presentation" hypothesis - the important and powerful idea that if you don't immediately *know* a fact, but you know how to derive that fact, you can re-present that fact to yourself whenever it is forgotten, and can thereby strengthen your direct recall memory for that fact over time.

Results from Experiments 3 and 4 supported this hypothesis. In those experiments, participants learned a set of artificial arithmetic facts, were tested on their ability to recall those facts,

learned a second distractor set of facts, then had the opportunity to answer problems from the first set of facts again, without any time pressure, but also without the benefit of any feedback, and finally were retested on their ability to rapidly recall the first set of facts. In both experiments, computation participants outperformed memorization participants during the feedback-less practice itself (presumably because they could compute what they could not remember), and even more notably, computation participants' performance improved from the first to the final speed test while memorization participants' performance declined.

Overall, this implies that computation may offer an important long-term benefit over rote memorization: when a learner who has practiced via self-computation forgets an answer, as is likely to happen over time, they can recreate that answer, re-exposing themself to the correct problem-answer association and, in turn, boosting their long term memory for that answer. By contrast, when a learner who has studied via rote memorization forgets the answer, they have no way to retrieve or recreate it and their memory for the studied facts worsens with time. This suggests that despite the fact that studying via rote memorization is initially more efficient, learning facts via self-computation may ultimately result in better long term memory.

Importantly, these results align with the intuition of a great many researchers and educators who have, for decades, posited that learning arithmetic facts in a meaningful, computationally-based way should be superior to learning them as a set of random associations to be memorized (e.g., Isaacs and Caroll, 1999; Baroody, Bajwa, and Eiland, 2009). However, despite the fact that intuitively it seems that this must be the case, the results of experimental studies have been mixed, with several failing to find any benefit of self-computation approach over rote memorization (e.g., Logan and Klapp, 1991). The present set of studies clarifies why that is. In all four studies, I show that when memory is tested immediately after study, computation appears less effective than memorization. Furthermore, even if memory is tested after a delay (Study 1, speed test 3), computation *still* appears less effective, so long as participants have not had an opportunity to continue practicing the studied

facts without feedback during that delay. Finally, and perhaps most importantly, even if participants have had an opportunity to continue practicing without feedback during the delay, if that feedback-less practice occurs when memory is already high (Study 2), both memorization and computation participants benefit from that practice, resulting in a persistent advantage for memorization participants. It is only in the particular case where participants have had the opportunity to *forget* a number of the learned facts, and *then* engage in practice without feedback that the advantage of computation practice emerges. Not only does this explain why this effect has proved so elusive in prior studies, but it is also, to my knowledge, the first clear mechanistic explanation of how computation practice might confer a long term advantage - with prior studies making reference to vague concepts such as "more interconnected knowledge" in students who learn via self-computation (Baroody, Bajwa, and Eiland, 2009).

6.3 Extensions of and Boundaries on the Self-re-presentation Effect

The results of this study would have a rather narrow application if we believed that they applied exclusively to arithmetic facts. Logically, however, any advantage that accrues to computation via this mechanism (the ability to self-re-present the answer when it is forgotten) should also apply to other "reconstructive methods," i.e., methods that allow a participant to re-derive the correct answer when it is forgotten, even if that method is not computation per se. For example, I might expect a similar effect to occur when I teach mnemonics such as the PEMDAS acronym for order of operations, when I teach general rules for spelling rather than teaching spelling words individually (i.e., "i before e except after c"), or when I teach students to use a finger trick to find multiples of nine. In general, I argue that arming students with methods that they can use to reconstruct information that is otherwise forgotten, should not only boost success when those reconstructive methods are used, but should also improve future recall.

This proposed mechanism also suggests some important boundaries on the effect. First, this only works if the reconstructive method is less likely to be forgotten than the target fact itself. I

expect this to occur in the following cases: (1) the reconstructive method is practiced more often than the target fact, or (2) the reconstructive method can itself be easily reconstructed from other well-known facts. Let's look at the first of these in the context of Experiments 3 and 4. Presumably the algorithm could itself be forgotten, and, if it were, then I should expect a computation participant to benefit no more from the feedback-less practice than a memorization participant. However, I found that very few participants appeared to have forgotten the algorithm. This is especially noteworthy in Study 4, where participants learned another, potentially interfering, algorithm during the delay. It is also important to note that clearly many of the individual facts learned by the memorization group *were* forgotten over the same delay. So why wasn't the algorithm forgotten? One simple explanation is that the algorithm was practiced more frequently than each individual fact. Over the course of responding to 48 items (six different problems with eight presentations of each) during the training, the algorithm was practiced up to 48 times (probably less, as some answers may have been directly retrieved on later trials), while each individual answer was practiced only 8 times. If this explanation holds, one implication for educators is that teaching fewer, more broadly applicable reconstructive methods may be better than teaching a variety of narrowly applicable reconstructive methods, because, in the latter case, those methods will be practiced less frequently and are themselves likely to be forgotten.

I just noted that one way to increase the likelihood that the reconstructive method is remembered is to ensure that the student has ample opportunities to practice that method. A second way to ensure that students remember the reconstructive method is to select a reconstructive method that can itself be reconstructed from other well known facts or methods. For example, suppose that students are taught to solve 5 x n (where n is any single digit number) via a complex series of steps that says: if the number is even, take half of the number and place a zero after that digit, and if the number is odd, subtract 1, then take half of the number, then place a 5 after that digit. Since this strategy is presented as a random collection of steps to be recalled, if a student

forgets any of the steps, they will have no way to recreate the strategy. On the other hand, if students are given a more transparent procedure, they should be able to reconstruct the procedure itself when it is forgotten. For example, suppose instead that students are reminded that every two fives make ten and so one needs to figure out how many groups of two can be made when multiplying by five and whether there will be any fives left over. For example, 7 x 5 can be thought of as seven fives. Every two fives makes a ten, so out of seven fives, take six of those fives and group them into three pairs, to make three tens, for a total of thirty. Of the original seven fives, one unpaired five will then be left over, yielding a final answer of 35. Such a procedure should be less prone to memory errors and bugs because students can always go back to first principles to derive the procedure if it is forgotten. This example also illustrates that reconstruction and self-re-presentation can occur on multiple nested levels - a student may use a computational algorithm to reconstruct individual facts and may also use other methods or facts to reconstruct the algorithm itself.

Relatedly, I noted anecdotally in Study 4, that some participants seemed to be using the few learned facts that they recalled to help reconstruct the algorithm and then using the algorithm to recompute other forgotten facts. This extends the idea above that reconstruction can occur on multiple levels, suggesting that it can also occur in mutually reinforcing feedback loops. This phenomenon is likely related to the finding that teaching participants multiple unrelated facts results in worse memory (as compared to learning a single fact), but teaching participants multiple *related* facts results in better memory (Bradshaw and Anderson, 1982), i.e., a redundancy effect. The key idea is that if people learn logically connected pieces of information, then when one piece of information is forgotten, other remembered pieces can be used to assist in recalling or inferring the forgotten piece. By contrast, when people learn *unrelated* facts, a forgotten fact cannot be inferred from the remembered ones.

As discussed above, one boundary on this effect is that students need to *remember* the algorithm or other reconstructive method. A second boundary on this effect is that it should only work if the student is capable of *correctly using* the reconstructive method to derive the forgotten fact. For example, if a participant in Study 3 or 4 frequently came up with the wrong answer when executing the algorithm, then I should expect that participant *not* to benefit from the feedback-less practice, and in general I should expect that participants' memory for the correct answers to decrease over time. Certainly in the limit where a student cannot execute the reconstructive method at all, it must be the case. Study 4 provides some suggestive (albeit correlational data) to support this; participants who performed poorly on the untimed test (8 participants who got less 50% correct, including 5 participants who got 0% correct) all failed to improve from the first to the third speed test, indicating that the additional practice during the untimed test did not help them. By contrast, of the 65 participants who performed well on the untimed test (>50 % correct), 48 improved in performance from the first to the third speed test. This suggests that learning a reconstructive method only benefits memory if that reconstructive method can be recalled and applied correctly.

Notably this ability to correctly execute the reconstructive method depends on both characteristics of the learners and characteristics of the algorithm. For example, it is possible that for certain students (e.g., perhaps students with certain learning disabilities), who struggle mightily to correctly execute the algorithm, it may be more effective to simply focus on rote memorization. Additionally, this theory suggests that the particular reconstructive method that we select should matter; selecting a reconstructive method that is difficult for students to execute may do more harm than good. For example, while it may be useful to teach students to solve 9 x n by doing 10 x n - n (e.g., 9 x 8 = 80 - 8), it may be less useful to teach them to solve 7 x n by doing 10 x n - 3 x n (e.g., 7 x 8 = 80 - (3 x 8)) since the latter is more error prone (due to involving an additional multiplication step, and requiring more difficult subtraction that crosses a decade), and may generate as many wrong answers as right ones, hurting long term memory for the answers. Relatedly, teaching students

"derived fact strategies," in which known facts are used to quickly compute unknown facts (e.g., 6 x n = 5 x n + n, or 9 x n = 10 x n - n, or 9 + n = 10 + n - 1) may prove more effective than teaching them counting strategies (e.g., solve 9 + 5 by counting on five counts, i.e, 10, 11, 12, 13, 14, or solve 8 x 6 by skip counting by eight six times, i.e., 8 ,16, 24, 30, 36, 42, 48), since counting strategies tend to be relatively error prone (Carpenter and Moser, 1984). For example, when counting by eight six times to solve 6 x 8 it is quite easy to say one number incorrectly and get off track or to lose count and count too few or too many eights. By contrast, there are a lot fewer ways to make a mistake when solving 6 x 8 by reasoning that 5 x 8 is 40 and 40 + 8 = 48. Importantly, however, as discussed above, students also need to *remember* the algorithm. We should be cautious about balancing competing benefits to strategy *memory* versus to accurate strategy *execution*. For example, if students are taught a single counting strategy that can be used to solve *all* multiplication problems (e.g., skip counting), they are very unlikely to forget the strategy itself. By contrast, teaching students different derived fact strategies for different facts (e.g., 6 x n = 5 x n + n, 9 x n = 10 x n - n, 4 x n = n x 2 x 2, etc.), may lead to fewer errors in executing each strategy, while simultaneously increasing the likelihood that students forget the strategies. These competing costs and benefits must be carefully weighed.

As another limitation, it is also important to note that my results suggest that not every fact that needs to be learned requires a reconstructive method. I would argue that if a fact is practiced enough, then rote memorization alone can be sufficient to produce robust long-term memory. For example people successfully learn the names of their friends and family, the location of items in a frequently visited store, the identity of the letters of the alphabet, and more, largely via repeated exposure rather than via any explicit reconstructive strategy. Supporting this conclusion, Study 2 demonstrated that if a fact is practiced frequently enough, then each retrieval of that fact will largely be successful and can serve to boost future memory for the fact, even without immediate corrective feedback. Indeed, this is a key lesson from comparing Study 2 to Studies 3 and 4. In Study 2, the

untimed test (feedback-less practice) occurred immediately after the first speed test. Participants essentially had no opportunity to forget the facts between the speed test and the untimed test. As a result both groups performed exceptionally well (~93% correct) on the untimed test, and both groups performed better on the subsequent speed test than they had on the first speed test. In Studies 3 and 4, there was a delay (in which participants learned a distractor set of facts) between the first speed test and the untimed test. By the time they reached the untimed test, participants in both conditions had likely forgotten many of the initially learned facts. Memorization participants then performed poorly on the untimed test, and their performance on the final speed test declined relative to the first speed test. Together these results suggest that memorization participants *can* benefit from feedback-less practice, if that practice occurs when their memory for the information is already high, i.e. if they largely experience retrieval successes rather than retrieval failures. It is only when they have forgotten much of the material that feedback-less practice becomes of little use to them.

Overall these results suggest that rote memorization can in fact be a viable alternative to teaching reconstructive methods, provided that we expect the learned material to be frequently practiced (i.e., practiced so often that little forgetting is expected to occur between exposures). For example, it is likely unnecessary to teach kindergarten students reconstructive methods to remember each of the 26 letters of the alphabet. They will have ample exposure to those letters over the course of their lives. By contrast, if the to-be-learned content will be practiced infrequently enough that forgetting will likely occur between exposures, then we should also provide students with methods that they can use to recreate those facts when they are forgotten. For example, it might be useful to provide students with reconstructive methods that they can use to re-derive the formulas for the area of various quadrilaterals (parallelograms, trapezoids, etc.) should they forget those formulas. Additionally it is possible that even for content that would eventually be mastered by rote memorization alone (e.g., certain spellings), adding reconstructive methods (e.g., learning rules like "i

before e except after c") can help speed that acquisition. This last point is an open empirical question.

6.4 Limitations of the Present Studies

The present set of studies also have several limitations, given that they were conducted under somewhat artificial circumstances (e.g. using adults and using artificial arithmetic). While they represent an important demonstration of a key memory mechanism, further work should be done with the actual target populations (elementary students) and target material (single digit arithmetic facts) to better understand how these mechanisms might play out in actual classroom settings. Additionally, the present set of studies tested only two specific learning methods. It remains entirely possible that another method might prove just as effective or more effective. For example, it is possible that providing students with a look-up table that they can use to answer questions if they need it might confer both some of the retrieval-practice benefits of flashcard-like memorization (because the time needed to scan the table would motivate them to first try to recall the answer if they can) and some of the re-presentation benefits of computation (because they can always look up an answer if they cannot recall it). Alternatively, it might make the most sense in real life to combine the two learning methods studied here, having students engage in both flashcard-like memorization and self-computation, again providing them with the benefits of retrieval-practice on direct recall ability and with the self-re-presentation benefits that accrue from being proficient a computation.

6.5 Concluding Remarks

In sum, this set of experiments helps to explain why, for nearly a century, educators and educational psychologists have often instinctively felt that self-computation must be superior to rote memorization, and yet, in controlled empirical studies, such effects have been elusive. I propose that self-computation practice results in poorer initial memory, but, can lead to superior long term memory as a result of the ability to self-re-present forgotten answers. That is, whenever a computation student cannot recall a particular answer, they can recompute it, and that re-exposure

to the correct answer will boost subsequent recall, resulting in a virtuous cycle that strengthens memory over time. I also note that this idea almost certainly applies to a broad class of material to be learned, not just arithmetic facts; in general, arming students with "reconstructive methods" that they can use to re-create forgotten material should have powerful positive effects on long term memory. For example, teaching general rules for spelling as opposed to having students separately memorize the spellings of individual words should boost long term memory via this mechanism. That said, I acknowledge that there are limitations to the situations in which it is worthwhile to teach reconstructive methods. Specifically, we should be careful to only teach reconstructive methods that are not themselves difficult to remember, e.g. due to being too numerous, too complicated, or too abstract. If the reconstructive method itself is forgotten, then it no longer confers any benefit to memory. Additionally, even if the reconstructive method is recalled, if it is difficult to actually execute, it should also offer very little benefit. Another important consideration is that reconstructive methods may not be necessary for some material that is practiced so frequently that little forgetting is expected to occur between exposures, as the real benefit of reconstructive methods only appears when material cannot be recalled. This nuanced answer provides a framework that can be applied to design effective classroom instruction and practice, as well as theory that can be explored future studies of human memory.

REFERENCES

Anwyl-Irvine, A. L., Massonnié, J., Flitton, A., Kirkham, N., & Evershed, J. K. (2020). Gorilla in our midst: An online behavioral experiment builder. *Behavior research methods*, 52(1), 388-407.

Bahrick, H. P., Bahrick, L. E., Bahrick, A. S., & Bahrick, P. E. (1993). Maintenance of foreign language vocabulary and the spacing effect. *Psychological Science*, 4(5), 316-321.

Baroody, A. J., Bajwa, N. P., & Eiland, M. (2009). Why can't Johnny remember the basic facts?. *Developmental disabilities research reviews*, 15(1), 69-79.

Baroody, A. J., Purpura, D. J., Eiland, M. D., & Reid, E. E. (2014). Fostering first graders' fluency with basic subtraction and larger addition combinations via computer-assisted instruction. *Cognition and Instruction*, 32(2), 159-197.

Baroody, A. J., Purpura, D. J., Eiland, M. D., & Reid, E. E. (2015). The impact of highly and minimally guided discovery instruction on promoting the learning of reasoning strategies for basic add-1 and doubles combinations. *Early Childhood Research Quarterly*, 30, 93-105.

Baroody, A. J., Purpura, D. J., Eiland, M. D., Reid, E. E., & Paliwal, V. (2016). Does fostering reasoning strategies for relatively difficult basic combinations promote transfer by K-3 students?. *Journal of Educational Psychology*, *108*(4), 576.

Batterson, J., (2012) *Beast Academy 3B guide.* AoPS Incorporated.

Bradshaw, G. L., & Anderson, J. R. (1982). Elaborative encoding as an explanation of levels of processing. *Journal of Verbal Learning and Verbal Behavior*, 21(2), 165-174.

Brownell, W. A. (1944). Rate, accuracy, and process in learning. *Journal of Educational Psychology*, 35(6), 321.

Brownell, W. A., & Chazal, C. B. (1935). The effects of premature drill in third-grade arithmetic. *The Journal of Educational Research*, 29(1), 17-28.

Campbell, J. I. (1987). The role of associative interference in learning and retrieving arithmetic facts. *Cognitive processes in mathematics*, 107-122.

Campbell, J. I. D. (1985). Associative interference in mental computation. PhD. Thesis, University of Waterloo, Ontario, Canada.

Campbell, J. I., & Graham, D. J. (1985). Mental multiplication skill: Structure, process, and acquisition. *Canadian Journal of Psychology/Revue canadienne de psychologie*, *39*(2), 338.

Carnine, D. W., & Stein, M. (1981). Organizational strategies and practice procedures for teaching basic facts. *Journal for Research in mathematics Education*, *12*(1), 65-69.

Carpenter, S. K., & DeLosh, E. L. (2006). Impoverished cue support enhances subsequent retention: Support for the elaborative retrieval explanation of the testing effect. *Memory & cognition*, 34(2), 268-276.

Carpenter, T. P., & Moser, J. M. (1984). The acquisition of addition and subtraction concepts in grades one through three. *Journal for research in Mathematics Education*, 179-202.

Cerella, J., Onyper, S. V., & Hoyer, W. J. (2006). The associative-memory basis of cognitive skill learning: Adult age differences. *Psychology and Aging*, *21*(3), 483.

Chard, D. J., Vaughn, S., & Tyler, B. J. (2002). A synthesis of research on effective interventions for building reading fluency with elementary students with learning disabilities. *Journal of learning disabilities*, 35(5), 386-406.

Cook, C. J., & Dossey, J. A. (1982). Basic fact thinking strategies for multiplication—revisited. *Journal for Research in Mathematics Education*, 13(3), 163-171.

Craik, F. I., & Tulving, E. (1975). Depth of processing and the retention of words in episodic memory. *Journal of experimental Psychology: general*, 104(3), 268.

Delazer, M., Ischebeck, A., Domahs, F., Zamarian, L., Koppelstaetter, F., Siedentopf, C. M., ... & Felber, S. (2005). Learning by strategies and learning by drill—evidence from an fMRI study. *Neuroimage*, *25*(3), 838-849.

De Visscher, A., & Noël, M. P. (2013). A case study of arithmetic facts dyscalculia caused by a hypersensitivity-to-interference in memory. *Cortex*, 49(1), 50-70.

De Visscher, A., & Noël, M. P. (2014). Arithmetic facts storage deficit: The hypersensitivity−to−interference in memory hypothesis. *Developmental science*, 17(3), 434-442.

De Visscher, A., Szmalec, A., Van Der Linden, L., & Noël, M. P. (2015). Serial-order learning impairment and hypersensitivity-to-interference in dyscalculia. *Cognition*, 144, 38-48.

De Visscher, A., & Noel, M. P. (2016). Similarity interference in learning and retrieving arithmetic facts. *Progress in brain research*, 227, 131-158.

Fuchs, L. S., Fuchs, D., Compton, D. L., Powell, S. R., Seethaler, P. M., Capizzi, A. M., ... & Fletcher, J. M. (2006). The cognitive correlates of third-grade skill in arithmetic, algorithmic computation, and arithmetic word problems. *Journal of Educational Psychology*, *98*(1), 29.

Fuchs, L. S., Powell, S. R., Seethaler, P. M., Cirino, P. T., Fletcher, J. M., Fuchs, D., & Hamlett, C. L. (2010). The effects of strategic counting instruction, with and without deliberate practice, on number combination skill among students with mathematics difficulties. *Learning and individual differences*, 20(2), 89-100.

Geary, D. C. (2011). Cognitive predictors of achievement growth in mathematics: a 5-year longitudinal study. *Developmental psychology*, 47(6), 1539.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Great Minds (2016). *Eureka Math: Grade 3 Module 3 Student Workbook v. 1.3.0*. Great Minds.

Haring, N.G., Lovitt, T.C., Eaton, M.D., & Hansen, C.L. (1978). *The fourth R: Research in the classroom*. Columbus, OH: Merrill.

Hasselbring, T. S., Goin, L. I., & Bransford, J. D. (1988). Developing math automatically in learning handicapped children: The role of computerized drill and practice. *Focus on exceptional children*, 20(6).

Henry, V. J., & Brown, R. S. (2008). First-grade basic facts: An investigation into teaching and learning of an accelerated, high-demand memorization standard. *Journal for Research in Mathematics Education*, 39(2), 153-183.

Hyde, T. S., & Jenkins, J. J. (1973). Recall for words as a function of semantic, graphic, and syntactic orienting tasks. *Journal of Verbal Learning and Verbal Behavior*, 12(5), 471-480.

Isaacs, A. C., & Carroll, W. M. (1999). Strategies for basic-facts instruction. *Teaching Children Mathematics*, 5(9), 508-515.

Kane, J. H., & Anderson, R. C. (1978). Depth of processing and interference effects in the learning and remembering of sentences. *Journal of Educational Psychology*, 70(4), 626.

Karpicke, J. D., & Blunt, J. R. (2011). Retrieval practice produces more learning than elaborative studying with concept mapping. *Science*, 331(6018), 772-775.

Karpicke, J. D., & Smith, M. A. (2012). Separate mnemonic effects of retrieval practice and elaborative encoding. *Journal of Memory and Language*, 67(1), 17-29.

Kaufman, J. H., Davis, J. S., Wang, E. L., Thompson, L. E., Pane, J. D., Pfrommer, K., & Harris, M. (2017). *Use of open educational resources in an era of common standards*. RAND Corporation, March, 27.

Keppel, G., & Underwood, B. J. (1962). Proactive inhibition in short-term retention of single items. *Journal of verbal learning and verbal behavior*, 1(3), 153-161.

Kornell, N., & Vaughn, K. E. (2016). How retrieval attempts affect learning: A review and synthesis. *Psychology of learning and motivation*, 65, 183-215.

Kuhn, M. R., & Stahl, S. A. (2003). Fluency: A review of developmental and remedial practices. *Journal of educational psychology*, 95(1), 3.

Lehman, M., Smith, M. A., & Karpicke, J. D. (2014). Toward an episodic context account of retrieval-based learning: Dissociating retrieval practice and elaboration. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 40(6), 1787.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological review*, *95*(4), 492.

Logan, G. D., & Klapp, S. T. (1991). Automatizing alphabet arithmetic: I. Is extended practice necessary to produce automaticity?. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*(2), 179.

McNamara, D. S. (1995). Effects of prior knowledge on the generation advantage: Calculators versus calculation to learn simple multiplication. *Journal of Educational Psychology*, *87*(2), 307.

Muth, C., Oravecz, Z., & Gabry, J. (2018). User-friendly Bayesian regression modeling: A tutorial with rstanarm and shinystan. *Quantitative Methods for Psychology*, 14(2), 99-119.

Myers, J. L., O'Brien, E. J., Balota, D. A., & Toyofuku, M. L. (1984). Memory search without interference: The role of integration. *Cognitive Psychology,* 16(2), 217-242.

National Council of Teachers of Mathematics. 2000. P*rinciples and standards for school mathematics: Standards 2000*. Reston, VA: National Council of Teachers of Mathematics.

National Mathematics Advisory Panel. (2008). *Foundations for success: The final report of the National Mathematics Advisory Panel*. Washington, DC: U.S. Department of Education.

National Research Council, & Mathematics Learning Study Committee. (2001). *Adding it up: Helping children learn mathematics*. National Academies Press.

Noël, M. P., & De Visscher, A. (2018). Hypersensitivity-to-Interference in Memory as a Possible Cause of Difficulty in Arithmetic Facts Storing. *Heterogeneity of Function in Numerical Cognition* (pp. 387-408). Academic Press.

Norem, G. M. (1928). The learning of the one hundred multiplication combinations. Unpublished master's thesis, State University of Iowa, Iowa City.

Osgood, C. E. (1949). The similarity paradox in human learning: A resolution. *Psychological review*, 56(3), 132.

Price, G. R., Mazzocco, M. M., & Ansari, D. (2013). Why mental arithmetic counts: brain activation during single digit arithmetic predicts high school math scores. *Journal of Neuroscience*, 33(1), 156-163.

Pyke, A., LeFevre, J. A., & Isaacs, R. (2008). Why Do The Math? The Impact of Calculator Use on Participants' Actual and Perceived Retention of Arithmetic Facts. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 30, No. 30).

Pyke, A. A., & LeFevre, J. A. (2011). Calculator use need not undermine direct-access ability: The roles of retrieval, calculation, and calculator use in the acquisition of arithmetic facts. *Journal of Educational Psychology*, *103*(3), 607.

Radvansky, G. A., & Zacks, R. T. (1991). Mental models and the fan effect. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 17(5), 940.

Rathmell, E. C. (1978). Using Thinking Strategies to Teach the Basic Facts. *NCTM Yearbook*, 13(38), 78.

Reder, L. M., & Ritter, F. E. (1992). What determines initial feeling of knowing? Familiarity with question terms, not with the answer. *Journal of Experimental Psychology: Learning, memory, and cognition*, 18(3), 435.

Rickard, T. C. (1997). Bending the power law: A CMPL theory of strategy shifts and the automatization of cognitive skills. *Journal of Experimental Psychology: General*, 126(3), 288.

Rittle-Johnson, B., & Kmicikewycz, A. O. (2008). When generating answers benefits arithmetic skill: The importance of prior knowledge. *Journal of Experimental Child Psychology*, *101*(1), 75-81.

Rohwer, W. D. (1966). Constraint, syntax and meaning in paired-associate learning. *Journal of Memory and Language*, 5(6), 541.

Rowland, C. A. (2014). The effect of testing versus restudy on retention: a meta-analytic review of the testing effect. *Psychological Bulletin*, *140*(6), 1432.

Siegler, R. S. (1988). Strategy choice procedures and the development of multiplication skill. *Journal of experimental psychology: General*, 117(3), 258.

Slamecka, N. J., & Graf, P. (1978). The generation effect: Delineation of a phenomenon. *Journal of Experimental Psychology: Human learning and Memory*, 4(6), 592.

Stein, B. S., & Bransford, J. D. (1979). Constraints on effective elaboration: Effects of precision and subject generation. *Journal of Verbal Learning and Verbal Behavior*, *18*(6), 769-777.

Steinberg, R. M. (1985). Instruction on derived facts strategies in addition and subtraction. *Journal for Research in Mathematics Education*, 16(5), 337-355.

Swenson, E. J. (1949). Organization and generalization as factors in learning, transfer, and retroactive inhibition. *Learning theory in school situations*, 9-39.

Thiele, C. L. (1938). *The contribution of generalization to the learning of the addition facts* (No. 763). Teachers College, Columbia University.

Thornton, C. A. (1978). Emphasizing thinking strategies in basic fact instruction. *Journal for Research in Mathematics Education*, 214-227.

Vasilyeva, M., Laski, E. V., & Shen, C. (2015). Computational fluency and strategy choice predict individual and cross-national differences in complex arithmetic. *Developmental Psychology*, 51(10), 1489.

Walker, D., Mickes, L., Bajic, D., Nailon, C. R., & Rickard, T. C. (2013). A test of two methods of arithmetic fluency training and implications for educational practice. *Journal of Applied Research in Memory and Cognition*, 2(1), 25-32.

Woodward, J. (2006). Developing automaticity in multiplication facts: Integrating strategy instruction with timed practice drills. *Learning Disability Quarterly*, *29*(4), 269-289.

APPENDICES

Appendix A. A Brief Overview of Bayesian Statistical Analyses

A.1. Overview

Classical frequentist statistics (i.e., the foundation for T-tests, ANOVA, traditional linear regression, etc.) relies on the null hypothesis test, which begins by *assuming* a particular state of the world (e.g., no difference between conditions) and then determines the probability of obtaining a result as extreme as the one actually observed (or more) *if* that null hypothesized state of the world were true. For example, suppose that a bag contains four balls, each of which might be black or white. Further suppose that we draw a single ball from the bag, observe its color, and then return it to the bag. We repeat this ten times and observe that eight times the ball is black and twice it is white. In frequentist statistics, we would analyze this result by saying, "assume that the bag contains equal numbers of black and white balls" (our null hypothesis), what is the probability of drawing a black ball on eight or more out of ten draws? This probability is our "p-value" (here $7/128 \approx 0.055$ or, "two-tailed" $14/128 = 0.11$, i.e if we consider eight or more white balls to be as surprising as eight or more black balls). These p-values suggest that it is not particularly unlikely to pull eight black balls and two whites out of a bag with equal numbers of blacks and white. Specifically, they say that if the bag really did have equal numbers of black and white balls, then 5.5% of the time (or 11% of the time two-tailed) we should expect a result this "surprising" by chance alone. We therefore do not reject the null hypothesis, i.e., we say that we do not have enough data to rule out the possibility that there are equal numbers of black and white balls in the bag.

However, in focusing solely on this one simple question: "how likely is our observed result if there are equal numbers of black and white balls in the bag?", we are throwing out a good deal of useful information. For example, although we cannot *rule out* the possibility that there are 2 black and 2 white balls in the bag, our data do suggest that it *is* likely that there are more black than white balls in the bag (since we drew eight blacks and only two whites). Frequentist statistics ignores this

entirely, and a result in which we obtained eight black balls and two whites is treated identically to a result in which we obtained five of each color, i.e., both are ruled "not significant". Or, more explicitly, in both cases, we say that the only conclusion that we can draw is that "we cannot reject the possibility that the bag contains equal numbers of black and white balls". This ignores a lot of valuable information in our data.

Bayesian statistics resolves this issue by considering not just *one* possible state of the world (e.g., are the two blacks and two whites in the bag or not?) but *all* possible states of the world. Specifically, we calculate how likely would we be to draw eight black balls and two white balls if there were no black balls in the bag? (The answer is that the probability is 0. As we might expect - it is impossible to draw eight black balls if there are no black balls.) How likely would we be to draw eight blacks and two whites and if there were one black and three whites in the bag? (The answer is that the probability is approximately 0.0004.) If there were two black balls? ($p \approx 0.04$) If there were three black balls? ($p \approx 0.28$) Or, finally, if there were four black balls? ($p = 0$). Notice that these probabilities are very similar to p-values and are calculated using the exact same laws of probability that generate p-values (with the small exception that now we consider *only* the specific case of drawing exactly eight black balls, i.e., what we actually observed, not the case of drawing eight *or more* black balls). The major difference from frequentist statistics is that instead of just doing the calculation once for a single possible state of the world (2 blacks and 2 whites in the bag), we do it five times for all possible states of the world (i.e., for 0, 1, 2, 3, or 4 black balls in the bag).

Bayesian statistics next compares these probabilities to determine how likely each of the underlying states of nature is based on our observations. For example, we can clearly see that it is impossible to observe eight black balls and two whites if there are zero or four black balls in the bag ($p = 0$), and that we are most likely to pull out eight black balls and two whites when there are three black balls ($p \approx 0.24$), with two black balls being second most likely ($p \approx 0.04$), and one black ball actually being quite unlikely given our results ($p \approx 0.0004$). We can flip this around to say that,

therefore, based on our data, it is most likely that the bag contains three black balls, second most likely that it contains two black balls, etc. If we want to be more precise, we can easily quantify this relative likelihood by simply dividing each of probabilities by the sum total of the fives probabilities, which basically serves to renormalize the values so that they sum to one. This gives us what are known as "posterior probabilities", here: 0 (0 black balls), 0.001 (1 black ball), 0.13 (2 black balls), 0.86 (3 black balls), 0 (4 black balls). These posterior probabilities can be understood as the probabilities that the bag contains the specified number of balls given that we observed eight black balls and two whites. For example, we can say that, based on having observed eight black balls on ten draws, there is a probability of 0.86 that the bag contains three black balls, a probability of 0.13 that it contains two black balls, and only a very tiny 0.001 probability that the bag contains only one black ball. This is much more informative than the results of a simple null hypothesis test.

A.2. Extension to the Case of a Continuous Parameter

The above toy example captures the general spirit of Bayesian inference, however it is worth mentioning two additional complicating factors. First, in my toy example the possible states of the world were discrete cases (i.e., a particular number of black balls), but more often they are continuous possibilities. For example, suppose that instead of drawing balls from a bag we were flipping a coin and trying to determine whether it was fair or biased (and if so, by how much). Further suppose that we observed eight heads on ten flips. We can imagine that we could do an identical series of calculations to those described above, with the complication that instead of only considering a small set of finite cases (i.e 0 , 1, 2, 3, or 4 black balls), we must consider an infinite number of cases, since the coin's probability of heads might take *any* continuous value between 0 and 1, i.e., we might ask how likely would we be to observe 8 heads and 2 tails if the true probability of heads were 0.1? 0.2? 0.3? 0.39? 0.7254? etc. The posterior in this case ends up being not a set of discrete probabilities but rather a probability distribution as shown in Figure A.1.

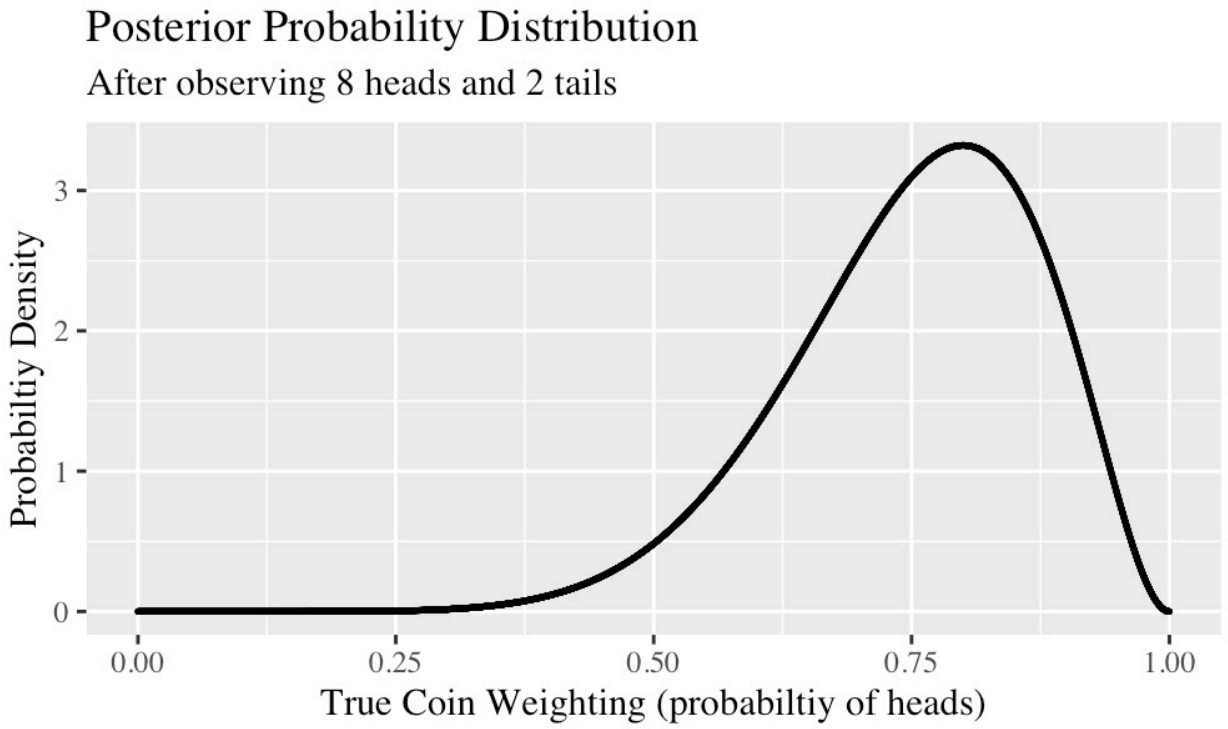

Figure A.1. Hypothetical posterior distribution for a coin's true probability of heads after having observed 4 heads and one tail.
For the technically curious, note that a Beta(1,1) prior was used in computing this distribution.

For example, according to Figure A.1, after observing eight heads and two tails, we can conclude that the most likely state of the world is that the coin is weighted more towards heads, with a probability of heads of 0.8 being the single most likely value (indicated by it having the highest probability density), though other probabilities in that neighborhood are also quite likely. In reporting the posterior, it is most common to actually display the entire posterior distribution (e.g., as in Figure A.1), although it is also common to compute some summary statistics to describe the distribution (I will return to this point below).

We can use Bayesian inference to estimate pretty much any unknown parameter value. So far in this chapter, I have used it to estimate the number of black balls in a bag and the weighting of a coin. In this dissertation, I use it to estimate the values of the parameters in a linear regression. For example, consider the simple linear regression: Child's IQ $= \alpha + \beta \cdot$ Parent's IQ. Typically, we

estimate the unknown parameters, α and β, using linear least squares, but it would also be easy to use Bayesian inference to quantify the probabilities that α and β take on particular values. In that case, you would end up with a poster distribution for α, i.e., showing all of the different possible values for α and their relative probabilities after having observed our data, and another posterior distribution for β as shown in Figure A.2. (For the more technically curious, note that these are actually the two marginal distributions from a joint posterior distribution over all possible combinations of α and β.)

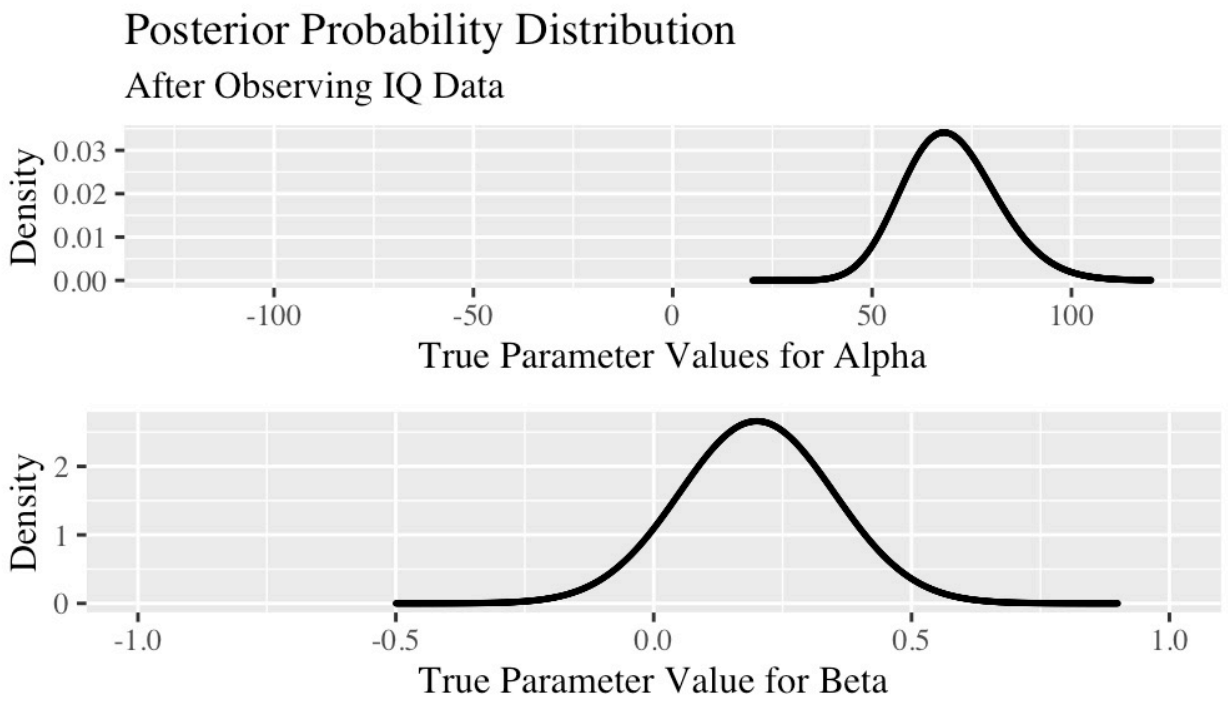

Figure A.2 Hypothetical posterior distributions for the intercept, alpha, and slope, beta. From a simple linear regression predicting child IQ from parent IQ. Note that the scale on the x-axis differs in the two plots.

As is typical of linear regression, we care most about the β parameter, which captures the relationship between parent and child IQ. How do we read this posterior? We can understand this posterior as showing that the most likely value of β is 0.2, which, if it were the actual true value, would mean that for every 1 point increase in parent IQ we should observe a 0.2 point increase in

child IQ. We also notice that a number of other possible values are also quite probable. In fact, looking at Figure A.2, we see that really anything in the range of 0 to 0.4 is pretty probable. That is, based on our data, we estimate that Child IQ increases anywhere from 0 to 0.4 points for every 1 point increase in parent IQ.

The posterior parameter distributions included in the body of this dissertation are analogous to the simple ones shown in Figure A.2. In addition to displaying the posterior parameter distributions, I further summarize my posterior distributions with three numbers. First, I report the distribution's median (favored by Bayesian statisticians over mean because it is less influenced by extreme values), although it is worth noting that for approximately normally distributed posteriors (which mine often are), the mean, median, and mode are roughly identical. We can therefore, understand the median as the mode, i.e as representing the most likely parameter value after having seen the data (e.g., 0.2 in the posterior shown above for the β parameter). I also report the posterior's median absolute deviation (MAD) - the median-based equivalent of standard deviation. This gives readers a sense of the spread of the distributions, i.e., is the posterior very tightly peaked around the median or is is widely spread around the median, making a broad range of parameter values quite probable? For example, for the β parameter shown in Figure 2.2, the median is 0.2 and the MAD is 0.15. As a general rule of thumb, for data that is approximately normal, the median is roughly equivalent to the mean, and the MAD is roughly equivalent to the SD, and approximately 95% of the posterior distribution falls within $\pm$ 2 MADs from the median. For example we can assume that approximately 95% of the posterior shown above falls in $0.2 \pm 2 \times 0.15$, i.e between -0.1 and 0.5. Put another way, there is only a 5% chance that the true value of the β parameter is less than -0.1 or more than 0.5. The reader is encouraged to apply this approximate "2 MADs" interpretation to the median and MAD values reported in the preceding chapters. Finally, because readers will be most familiar with frequentist statistics and p-values, I provide "Bayesian p-values", giving the probability that the true parameter has the opposite sign from the sign of the median value. For example, in the

case of the IQ data, we find that it is most likely that there is a positive relationship between parent IQ and child IQ (since our median value for β is a positive number, 0.2). I then ask, according to the posterior distribution, what is the probability that the true value of β is actually negative? This is computed by calculating the area under the posterior distribution curve to the left of zero (i.e., from negative infinity to zero). In the case of the IQ example, it is 0.09, meaning that, after having observed our data, we conclude that there is a 9% chance that the true relationship between Parent IQ and Child IQ is negative. While these values have a different interpretation than frequentist p-values, they may be comforting to some readers who are used to thinking in terms of binary tests, i.e., is there a positive relationship or not? They are also very easy to interpret, as they literally represent the stated probability. That is, you can directly understand a Bayesian p-value of 0.09 as meaning that there is a 9% chance that the true effect of ParentIQ on ChildIQ is zero or negative.

A.3. A Word About Priors

Finally, for completeness, I should also mention a second complicating factor that makes real-world Bayesian analysis slightly more complicated than my toy example with the black balls. The calculations of the posterior probabilities in my toy example was based on assuming that, *a priori*, i.e., prior to seeing the data, we believed that all possible states of the world were equally likely. That is, prior to pulling any balls out of the bag, we believed that it was equally likely that the bag contained 0 black balls, 1 black ball, 2 black balls, etc. We refer to this as having a "flat prior". If this is not the case, there is simply one additional step in the calculation in which the likelihood is multiplied by the prior probability before dividing by the sum total to renormalize. This allows us to easily incorporate prior beliefs into our calculation. For example, if we had *previously* drawn a three balls from the bag, i.e., before drawing the ten balls in our "experiment", and had observed that all were black, then on the basis of that information we should already suspect that there are more black than white balls in the bag. We can incorporate the information into our calculation in the form of a prior, which would weight the possibilities with more black balls more heavily than the possibilities with more

white balls. Or if we knew that whoever filled the bag was, in fact, trying to put two black and two white balls in, and any other result would have occurred only via human error, then we should assign a very high prior probability to the possibility that there are 2 black balls in the bag, a relatively low prior probability to the possibilities that there are 1 or 3 black balls in the bag, and a near zero prior probability to the possibility that there are 0 or 4 black balls in the bag (that would be quite the human error!). Again, mathematically, priors are easily incorporated into our calculation, as we simply multiply each calculated probability by the prior probability before dividing all probabilities by their sum to renormalize them.

Priors tend to make researchers who practice frequentist statistics nervous, as they worry that Bayesian researchers will pick any prior that they fancy, thereby biasing the results. One way around this is to use a flat prior, thereby introducing no bias. (Indeed the Bayesian estimates obtained with a flat prior are identical to the estimates obtained using maximum-likelihood estimation.) However in practice, it is actually more useful to employ very weak priors (as opposed to flat priors), typically in the form of a *very* wide normal distribution centered on zero. This keeps the computational algorithm that fits the model from considering exceptionally large or small values of the parameters, improving computational efficiency. For example, there is no need to consider the possibility that $\beta = 500$, i.e., that for each one point increase in parent IQ there is a 500 point increase in child IQ. Such a relationship is basically impossible. However, if we used a flat prior, the computer would consider this, as well as many other essentially impossible values for β. When we specify a flat prior, we tell the computer that *every* possible value of β (from negative infinity to infinity) is equally likely. In practice that is just silly, not to mention computationally wasteful. This is why we commonly use very weak priors in place of flat priors - they help the computer consider only reasonable values for the parameters.

Two additional points about priors are worth noting. First, because the priors are very weak, i.e., very, very wide normal distributions, the posterior tends to almost entirely reflect the influence

of the data, showing hardly any contribution of the prior at all. Second, because the priors are normal distributions centered on zero, to the extent that they do influence the posterior at all, they tend to pull the estimates towards zero, i.e., making the results more conservative and the researchers *more* likely to conclude that there is *no* effect. Thus, contrary to the popular fear that Bayesian models allow researchers to invent non-existent effects, Bayesian models are actually less likely to generate false positives than traditional frequentist models (Gelman et al., 2013).

A.4. Details About My Models

All of the models reported in preceding chapters were fit using the rstanarm Package in R (see Muth, Oravecz, and Gabry, 2018), which is convenient wrapper for Stan, a method of fitting Bayesian models using random sampling from the posterior obtained via Hamiltonian Monte Carlo. It should be noted that I did not select my own priors, but rather used the defaults included in this package, which have been carefully selected by Bayesian statisticians to reflect current best practice, and are, as described above, very wide normal distributions centered on zero.

It is also worth noting that my models are logistic regressions (because I am predicting a binary outcome, i.e., each response is either correct or incorrect). This is a separate issue from them being Bayesian. That is, like linear regression, logistic regression can also be fit using traditional frequentist statistics. The reader should, keep the logistic nature of the regression in mind as they attempt to interpret the regression coefficients, as coefficients have a much less straightforward interpretation in logistic regression. For example, suppose that parent IQ were instead used to predict a binary outcome: whether or not a child graduated from college. Our model might be:

$$logit(p_{\text{graduate}}) = \alpha + \beta \cdot \text{Parent IQ}$$

Which means:

$$log\left(\frac{p_{\text{graduate}}}{1 - p_{\text{graduate}}}\right) = \alpha + \beta \cdot \text{Parent IQ}$$

Or, by algebra:

$$p_{\text{graduate}} = \frac{e^{\alpha+\beta\cdot\text{Parent IQ}}}{1 + e^{\alpha+\beta\cdot\text{Parent IQ}}}$$

It is clear that it is no longer a simple case of saying that a 1 unit increase in Parent IQ results in a β unit increase in the outcome (probability of graduating). This relationship becomes even more complicated when there are multiple predictors in the model. However, it is still always the case that a positive coefficient, i.e., a positive β, indicates that increasing the predictor increases the outcome (e.g., the higher the parent's IQ the greater the probability that the child graduates), and a negative coefficient indicates that increasing the predictor decreases the outcome. This makes it easy to at least interpret the direction of the effect if not the precise magnitude of the effect. In cases where it is important to also understand the magnitude of the effect, I provide posterior predictions, i.e., the range of likely outcomes that the model would actually predict, to make the interpretation more transparent.

Appendix B. List of Typing Items.

The same thirty typing items were used in the typing screener in all four studies. They were (in order):

Item #	Three Digit Number to Be Typed
1	470
2	861
3	984
4	881
5	586
6	864
7	107
8	837
9	897
10	364
11	282
12	368
13	793
14	331
15	535
16	105
17	165
18	348
19	758
20	973
21	375
22	197
23	153
24	235
25	814
26	268
27	923
28	461
29	136
30	288

Appendix C. List of Multiplication Items.

There were two versions of the multiplication pre-test: a long version consisting of all permutations of the digits 2 through 9 (64 items total), and a short version, which was identical to the long but omitted all problems with a 2 or 5 as one of the multiplicands (36 items total). Both versions opened with three additional easy warm-up items. The items were (in order):

Studies 1 & 4 (Short Version)			
1	4	x	10
2	1	x	4
3	10	x	5
4	6	x	4
5	9	x	9
6	3	x	7
7	4	x	4
8	3	x	9
9	8	x	6
10	6	x	7
11	7	x	3
12	7	x	7
13	8	x	3
14	7	x	6
15	9	x	3
16	7	x	8
17	3	x	4
18	6	x	9
19	3	x	6
20	9	x	8
21	4	x	8
22	6	x	8
23	7	x	4
24	6	x	6
25	8	x	4
26	9	x	6
27	4	x	3
28	3	x	3
29	3	x	8
30	9	x	7
31	4	x	7

Studies 2 & 3 (Long Version)			
1	4	x	10
2	1	x	4
3	10	x	5
4	6	x	4
5	9	x	9
6	2	x	4
7	3	x	7
8	4	x	5
9	7	x	5
10	7	x	2
11	4	x	4
12	3	x	9
13	8	x	6
14	6	x	7
15	2	x	5
16	5	x	4
17	5	x	6
18	7	x	3
19	8	x	2
20	7	x	7
21	8	x	3
22	7	x	6
23	5	x	7
24	2	x	3
25	5	x	8
26	9	x	3
27	7	x	8
28	3	x	5
29	2	x	7
30	3	x	4
31	9	x	5

(continued on next page)

Item #	Studies 1 & 4 (cont.)		
32	6	x	3
33	9	x	4
34	8	x	7
35	4	x	6
36	8	x	9
37	7	x	9
38	4	x	9
39	8	x	8

Item #	Studies 2 & 3 (cont.)		
32	6	x	5
33	6	x	9
34	5	x	5
35	3	x	6
36	5	x	3
37	9	x	8
38	4	x	8
39	6	x	8
40	7	x	4
41	2	x	6
42	5	x	2
43	6	x	6
44	2	x	8
45	9	x	2
46	8	x	4
47	5	x	9
48	9	x	6
49	4	x	3
50	3	x	3
51	3	x	8
52	4	x	2
53	9	x	7
54	4	x	7
55	6	x	3
56	9	x	4
57	8	x	7
58	8	x	5
59	4	x	6
60	8	x	9
61	2	x	2
62	7	x	9
63	6	x	2
64	4	x	9
65	2	x	9
66	8	x	8
67	3	x	2

Appendix D. Delay Test Given to Memorization Participants in Study 4

In Study 4, memorization participants competed a short, computer paced test of arithmetic facts. This was included to account for the additional time and cognitive demands that were required of the computation participants as they were introduced to the new algorithm via a set of four example problems. Each item on the delay test was based on an example problem given to the computation participants to ensure that participants in both conditions were engaging in similar cognitive operations. Notice that problems were intentionally presented in scrambled order, some problems were presented in the reverse order (e.g., 8 - 6 = 2 instead of 2 + 6 = 8), and occasionally the fact itself was slightly modified (e.g., 3 x 60 instead of 6 x 30). This was all done to ensure that no memorization participant suspected that these problems somehow formed part of an algorithm that could be used to solve the a ~ b problems. The questions were (in order):

Memorization Delay Test Item #	Memorization Delay Test Question			Corresponding Example Problem in Computation Training*
1	5	x	30	4 ~ 5 = (5 x 30) - (4 + 5) = 150 - 9 = 141
2	4	+	5	4 ~ 5 = (5 x 30) - (4 + 5) = 150 - 9 = 141
3	5	+	6	5 ~ 6 = (6 x 30) - (5 + 6) = 180 - 11 = 169
4	3	x	60	2 ~ 6 = (6 x 30) - (2 + 6) = 180 - 8 = 172
5	5	+	3	5 ~ 3 = (3 x 30) - (3 + 5) = 90 - 8 = 82
6	180	-	11	5 ~ 6 = (6 x 30) - (5 + 6) = 180 - 11 = 169
7	90	/	3	5 ~ 3 = (3 x 30) - (3 + 5) = 90 - 8 = 82
8	82	+	8	5 ~ 3 = (3 x 30) - (3 + 5) = 90 - 8 = 82
9	180	-	8	2 ~ 6 = (6 x 30) - (2 + 6) = 180 - 8 = 172
10	60	x	3	5 ~ 6 = (6 x 30) - (5 + 6) = 180 - 11 = 169
11	150	-	9	4 ~ 5 = (5 x 30) - (4 + 5) = 150 - 9 = 141
12	8	-	6	2 ~ 6 = (6 x 30) - (2 + 6) = 180 - 8 = 172
13	9	-	4	4 ~ 5 = (5 x 30) - (4 + 5) = 150 - 9 = 141
14	3	x	30	5 ~ 3 = (3 x 30) - (3 + 5) = 90 - 8 = 82
15	11	-	5	5 ~ 6 = (6 x 30) - (5 + 6) = 180 - 11 = 169

*Note that there were only 4 distinct computation training problems. They each appear multiple times in the table because they were intentionally chopped up into pieces and scattered randomly throughout the memorization delay test.

Appendix E. Follow-Up Questions Asked at the Conclusion of Each Study.

<table>
<tr><td>Study 1
(Supervised)</td><td>Study 2
(Unsupervised)</td><td>Study 3
(Unsupervised)</td><td>Study 4
(Supervised)</td></tr>
<tr><td colspan="4">As you were learning the problems today, did you notice any patterns in the answers or any relationship between the numbers in each problem and the answer to that problem? Explain.</td></tr>
<tr><td colspan="4">Describe your experience learning the problems today. Any tricks or strategies that you used? Anything interesting that you noticed about how you learned or how you performed on the tests?</td></tr>
<tr><td></td><td colspan="3">Instructions Check. In today's experiment:
All keyboards have number keys at the top. Some keyboards have numeric keypads (i.e., numbers in a square) on the right side. You were supposed to use _____.
• the numbers keys at the top
• the number keys in the numeric keypad on the right side
• either of the above
You _____ allowed to use a calculator.
• were
• were not
You were supposed to type all responses with _____.
• your left hand
• your right hand
• both hands
You _____ allowed to use scratch paper.
• were
• were not</td></tr>
<tr><td></td><td></td><td>NOTE: Your honesty in responding to the next two questions (#4 and #5) is very important for our data. Your answers to these items will NOT affect your payment.</td><td></td></tr>
<tr><td></td><td></td><td colspan="2">Did you follow all of the instructions above during the experiment (e.g., about the keyboard, calculator, hands, and scratch paper)?
• Yes - I followed all instructions
• No - I did NOT follow the instructions
If you did not follow the instructions, please provide more detail on how specifically you deviated from the instructions:</td></tr>
<tr><td></td><td colspan="2">Tell us about the quality of the data you provided.
Excellent: No distractions, followed all of the instructions, gave my best effort.
Good: 1 or 2 minor distractions lasting <10 seconds each (e.g., phone rang but I silenced it), gave >90% effort.
Fair: Distractions totaling 10-60 seconds (e.g., had a brief conversation with someone who walked in), and/or gave 60%-90% effort.
Poor: Distractions totaling more than 1 minute (e.g., had the TV on throughout, had a whole conversation via text), and/or gave minimal effort.
Rate the quality of your data:
• Poor
• Fair
• Good
• Excellent</td><td></td></tr>
<tr><td colspan="4">Is there anything else you would like to share that would help us in analyzing your data or in learning more about human learning?</td></tr>
<tr><td colspan="4">Enter your age:</td></tr>
<tr><td colspan="4">Select your gender:
• male
• female
• non-binary</td></tr>
<tr><td></td><td></td><td>I certify that all of my answers are true to the best of my knowledge. I understand that my responses to #4 and #5 will not affect my payment.</td><td></td></tr>
</table>

To read the table, look at the questions that appear in each study column. For example, the first two questions (about noticing patterns and about describing your experience) appeared in all four studies. The instructions check question appeared in Studies 2-4 only. The note about honesty appeared in Study 3 only. And so on.

Appendix F. Exclusion Criteria and Analyses with modified Exclusion Criteria

Sample Characteristics and Exclusion Criteria

<table>
<tr><th>Study</th><th>Supervised</th><th>Source</th><th>Compensation</th><th>Automated In-Study Exclusion Criteria</th><th>Post-Completion Exclusion Criteria (Excluded from All Analyses)</th><th>Additional Exclusion Criteria (Excluded from Restricted Sample)</th></tr>
<tr><td>1</td><td>Yes</td><td>Prolific</td><td>Cash</td><td>Typing < 70%
Multiplication < 40%</td><td rowspan="4">All Studies:
• Unstable internet connection resulting in timing issues
• Technical glitches resulting in study delivery problems or missing data

Supervised Studies:
• Completed without supervision

Unsupervised Studies:
• Failed instructions check
• Self-reported not following instructions
• Self-reported “poor” data quality
• Did not complete in one sitting</td><td rowspan="4">All Studies:
• Typing < 70%

Unsupervised Studies:
• Self-reported “Fair” data quality</td></tr>
<tr><td>2</td><td>No</td><td>Sona</td><td>Credit</td><td>Typing < 20%
Multiplication < 20%</td></tr>
<tr><td>3</td><td>No</td><td>Prolific</td><td>Cash</td><td>Typing < 50%
Multiplication < 40%</td></tr>
<tr><td>4</td><td>Yes</td><td>Sona</td><td>Credit</td><td>None</td></tr>
</table>

Table F1. Sample Characteristics and Exclusion Criteria

Explanation of automated in-study exclusion criteria:

Automated in-study exclusion criteria varied from study to study (see table above). This warrants some explanation.

1. Study 2 came first chronologically (see footnote in Chapter 3). The participants were unsupervised and were compensated with course credit. As participants were unsupervised, the goal of the 20% typing and multiplication exclusion thresholds was simply to screen out participants who were not attending to the task. (It turned out that everyone met or exceeded these minimums.)

2. Study 3 came second chronologically. This sample was recruited from Prolific and I was worried that the quality of the participants might be lower than I was used to at my home university (i.e., in Study 2 and in earlier in-person studies). Poor typing skills would be a concern because they would limit a participant's ability to demonstrate their knowledge on the speed tests. Poor multiplication skills would be a concern, because in earlier in-person studies, I had observed that participants with very poor multiplication skills, who ended up being assigned to the computation condition, often struggled to complete the training in the allotted time. I intentionally set my typing and multiplication cutoffs to ensure that the Prolific sample was comparable to Study 2's Sona sample. The threshold for the typing measure was set at 50% because in Study 2, all but two participants (out of 81) scored above this threshold. Similarly, the

threshold for the multiplication measure was set 40% because in Study 2, all but two participants scored above that threshold.

3. Study 1 came third chronologically. In looking at the Study 2 and 3 data, I noticed that among participants who scored above approximately 70% on the typing measure, there was a great deal of variance in the performance on the speed tests. Among participants who scored below 70% on the typing measure, performance on the speed tests was universally low. This led me to believe that the poor typing skills of these participants were greatly limiting their ability to demonstrate their knowledge on the speed tests. I chose to exclude them in Study 1. The multiplication cut off remained at 40% as again, the primary concerns here was only that participants with very poor multiplication skills might not complete the study in the allotted time.

4. Study 4 came fourth chronologically. I elected not to screen these participants. They participated for course credit (unlike Studies 2 and 3 which were for cash), and it was impossible to award credits in increments of less than 0.5 units (equivalent to 30 minutes of participation). Since the screening measures took less than 10 minutes to complete, I wouldn't have known how to compensate participants who made a good faith effort to participate but were rejected for failing these screeners at the 70% and 40% thresholds. Awarding no credit seemed unfair. Awarding 0.5 credits for only 10 minutes of participation seemed like it would incentivize poor performance. I also didn't bother with the 20% thresholds from Study 2, which were intended as attention checks in that unsupervised study. They seemed unnecessary in this study given that (a) no one failed the attention checks in Study 2 using a sample drawn from the same population, and (b) I was watching the participants via Zoom in this study and could easily assess attention. I *did* fully intend that when actually analyzing my data, I would exclude the participants using the same criteria from Study 1. That is, I expected to collect full data from everyone (due to the credit issue), but only planned to *analyze* the data from participants who scored at or above 70% on the typing measure and 40% on the multiplication measure. However, as I neglect to mention that intention in my pre-registration, I report the data from *all* participants in the text (i.e., a comparable sample to Study 2), and use the Study 1 rejection criteria in the restricted sample reported below.

Explanation of Post Completion Rejection Criteria:

No justification is needed for these. They are all standard and obvious with the exception of rejecting participants for self-reporting "poor" quality data, which was explicitly listed in my pre-registrations as a rejection criterion.

Explanation of Additional Rejection Criteria (for the Restricted Sample):

For the restricted sample, the decision to exclude participants with <70% on the typing measures follows directly from the explanation for Study 1 above. I strongly believe that data from these participants is unreliable, as their poor typing skills limit their ability to demonstrate their knowledge on the speed tests.

I did not exclude additional participants on the basis of their multiplication scores. If they can type fine, but struggle with multiplication, I am not concerned about any measurement validity issues. The multiplication cut-offs were included in the in-study screening measures primarily because I had observed that participants with very poor multiplication skills were typically unable to complete the

computation condition in the allotted time. That is not something that makes sense to address after the fact.

The decision to exclude "fair" participants from the restricted sample follows directly from my pre-registrations, which stated that analyses would be run with and without these participants.

Summary of Omitted Participants

Study	Enrolled	Voluntarily Withdrew	Rejected	Completed the Study
1	310	128 Prior to Assignment* 0 Memorization 0 Computation	51 Failed Typing 42 Failed Multiplication	89
2	92	8 Prior to Assignment 1 Memorization 0 Computation	0 Failed Typing 0 Failed Multiplication 2 Technical Issue, Incomplete Data	81
3	85	2 Prior to Assignment 1 Memorization 2 Computation	9 Failed Typing 7 Failed Multiplication	64
4	160	17 Prior to Assignment 0 Memorization 0 Computation	1 Very Poor Internet Connection	142

Table F2. Participants who did not Complete the Study

* Recall that Study 1 was the only two-part study (i.e., with the multiplication and typing screener administered on a separate day from the rest of the study). It has much higher attrition than the others due largely to participants not returning for the second part. A significant number of participants also withdrew early on in the screener. It appears that they did not carefully read the description in Prolific that stated that they would need to agree to a video call for Part II. This was re-emphasized at the beginning of the screener, and a number of participants withdrew at that point.

Study	Completed the Study	Excluded from <u>all</u> Analyses	Final Sample (reported in text)	Excluded from Restricted Sample	Restricted Sample (reported below)
1	89	2 completed without supervision 1 unstable internet connection, timing issues	86 43 Mem 43 Comp	None	N/A
2	81	1 technical issue, incorrect # of trials 1 didn't complete in one sitting 1 failed instructions check	78 39 Mem 39 Comp	7 "fair" quality data 12 typing < 70%	59 31 Mem 28 Comp
3	64	1 failed instructions check 2 reported not following instructions	61 33 Mem 28 Comp	4 "fair" quality data 7 typing <70%	50 27 Mem 23 Comp
4	142	1 technical issue, incorrect # of trials 2 completed without supervision	139 66 Mem 73 Comp	12 typing < 70%	127 63 Mem 64 Comp

Table F3. Participants who Completed the Study but were Excluded from Analyses

Comparison of Final Sample and Restricted Results

Study	Data Set	Parameter	Final Sample Estimates (Reported in the Body of the Dissertation)			Restricted Sample Estimates		
			Median	MAD	$\begin{cases}\beta > 0 : P(\beta \leq 0) \\ \beta < 0 : P(\beta \geq 0)\end{cases}$	Median	MAD	$\begin{cases}\beta > 0 : P(\beta \leq 0) \\ \beta < 0 : P(\beta \geq 0)\end{cases}$
2	Test 1	Learn Condition Memorization (β_{LC})	1.04	0.33	0.001	1.11	0.41	0.005
	Test 2		0.75	0.36	0.02	0.73	0.44	0.04
	Untimed Test		0.10	0.55	0.43	0.11	0.62	0.43
	Tests 1 & 2	Learn Condition Memorization x Task Test 2 ($\beta_{LC\times TR}$)	-0.27	0.24	0.13	-0.37	0.28	0.10
3	Test 1	Learn Condition Memorization (β_{LC})	0.92	0.48	0.03	0.80	0.50	0.05
	Test 2		-0.60	0.52	0.13	-0.42	0.59	0.24
	Untimed Test		-3.35	0.80	<0.0001	-2.76	0.79	0.0005
	Tests 1 & 2	Learn Condition Memorization x Task Test 2 ($\beta_{LC\times TR}$)	-1.44	0.34	<0.0001	-1.28	0.36	0.0002
4	Test 1	Learn Condition Memorization (β_{LC})	0.56	0.32	0.05	0.69	0.34	0.03
	Test 2		0.43	0.30	0.08	0.45	0.30	0.08
	Test 3		-1.92	0.49	0.0001	-1.98	0.53	<0.0001
	Untimed Test		-4.41	0.70	<0.0001	-4.72	0.77	<0.0001
	Tests 1 & 2	Learn Condition Memorization x Task Test 2 ($\beta_{LC\times TR}$)	-0.11	0.27	0.35	-0.18	0.28	0.26
	Tests 1 & 3		-2.41	0.37	<0.0001	-2.50	0.37	<0.0001

Table F4. Posterior Parameter Estimates for the Full and Restricted Samples

Overall, there are no differences in the conclusions that would be drawn from the restricted sample as opposed to the final sample.