Title
Cognitive Models for Abacus Gesture Learning

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 46(0)

Authors
He, Lingyun
Ka, Duk Hee
Ehtesham-Ul-Haque, Md
et al.

Publication Date
2024

Peer reviewed

Cognitive Models for Abacus Gesture Learning

Lingyun He
(iris.he@psu.edu)
Department of Industrial and Manufacturing Engineering,
The Pennsylvania State University
University Park, PA 16802 USA

Duk Hee Ka
(dukhee.ka@psu.edu)
Department of Industrial and Manufacturing Engineering,
The Pennsylvania State University
University Park, PA 16802 USA

Md Ehtesham-Ul-Haque
(mfe5232 @psu.edu)
College of Information Science and Technology,
The Pennsylvania State University
University Park, PA 16802 USA

Syed M. Billah (sbillah@psu.edu)
College of Information Science and Technology
The Pennsylvania State University
University Park, PA 16802 USA

Farnaz Tehranchi (farnaz.tehranchi@psu.edu)
School of Engineering Design and Innovation
The Pennsylvania State University
University Park, PA 16802 USA

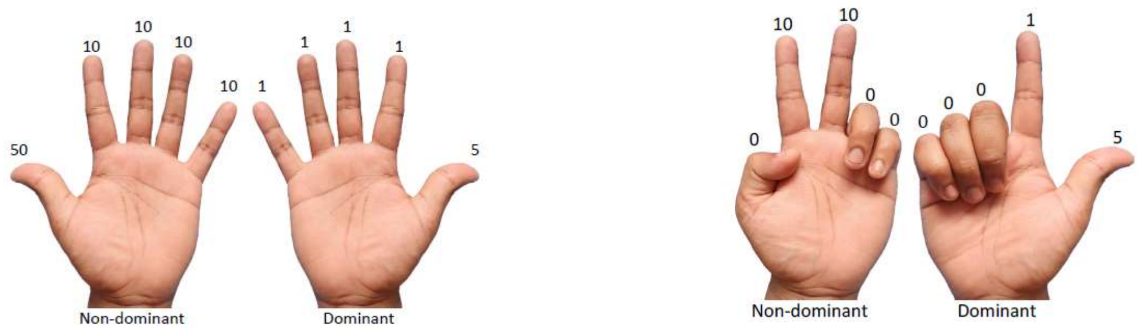

Figure 1. Illustration of Abacus Gestures (Ehtesham-Ul-Haque & Billah, 2023). *Left*: The set of numbers assigned to the 10 fingers in abacus gestures. On the non-dominant hand, each finger represents a value of 10, while the thumb represents a value of 50. On the dominant hand, each finger represents a value of 1, while the thumb represents a value of 5. *Right*: An example of an abacus gesture representing the number 26, formed by opening the index and middle fingers of the non-dominant hand and the thumb and index finger of the dominant hand.

Abstract

In this paper, we developed three ACT-R cognitive models to simulate the learning process of abacus gestures. Abacus gestures are mid-air gestures, each representing a number between 0 and 99. Our models learn to predict the response time of making an abacus gesture. We found the accuracy of a model's predictions depends on the structure of its declarative memory. A model with 100 chunks cannot simulate human response, whereas models using fewer chunks can, as segmenting chunks increase both the frequency and recency of information retrieval. Furthermore, our findings suggest that the mind is more likely to represent abacus gestures by dividing attention between two hands rather than memorizing and outputting all gestures directly. These insights have important implications for future research in cognitive science and human-computer interaction, particularly in developing vision and motor modules for mental states in existing cognitive architectures and designing intuitive and efficient mid-air gesture interfaces.

Keywords: Finger counting, abacus gesture, mid-air interaction, cognitive model, ACT-R, cognitive architectures.

Introduction

Abacus gestures are a large set of bare-hand gestures (Cho & So) that can be easily performed by users through opening and closing their 10 fingers (Ehtesham-Ul-Haque & Billah, 2023). Each gesture in this set corresponds to a numeric value between 0 and 99, with each finger having a specific worth. On the dominant hand, when opened, the thumb is worth 5, and each of the other four fingers is worth 1 (Figure 1). On the non-dominant hand, when opened, the thumb is worth 50, and each of the other fingers is worth 10. A finger is worth 0 when closed. Finger abacus has its root in East Asian culture and is used to perform basic arithmetic operations rapidly by counting fingers (Pai et al., 1981).

In the field of Human-Computer Interaction (HCI), abacus gestures are broadly categorized as mid-air gestures that can facilitate mid-air interaction. Our prior work has shown that abacus gestures are not only easy to learn but also easily detectable by 2D commodity cameras, such as webcams, using readily available computer vision packages (Ehtesham-Ul-Haque & Billah, 2023). As we transition from the post-PC and post-mobile era to virtual reality and spatial computing, abacus gestures have the potential to be utilized in a wide range of applications, such as target acquisition, alphanumeric symbol input, and command issuing.

Although empirical evidence has shown that abacus gestures are easy to learn, it is currently unknown why and how individuals process them in their minds. This paper aims to fill this gap. Specifically, we aim to understand the following research questions:

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

(1) Can a cognitive model predict human mind behavior when using abacus gestures to represent the numbers 0-99?

(2) How accurate are the predictions made by cognitive models?

(3) What factors contribute to the process of learning abacus gestures?

To address these questions, we first utilized human-subject data on the time taken to make abacus gestures from our prior work (Ehtesham-Ul-Haque & Billah, 2023) as ground truth. Next, we created three different cognitive models using Adaptive Control of Thought-Rational (ACT-R) (Anderson et al., 1997; Ritter et al., 2018, 2019) and compared their predicted response times for making each abacus gesture to those of the ground truth. Our findings strongly suggest that it is possible to create cognitive models that mimic how humans learn to make abacus gestures. Such models must consider the dominant and non-dominant hands separately and utilize a smaller number of chunks. Thus, we contribute to cognitive science by providing insights into the mental processes involved in learning and performing abacus gestures, which can inform the design of more intuitive and efficient mid-air interaction techniques in HCI.

Cognitive Model & ACT-R

Cognitive models are designed to predict individuals' behavior and provide insights into how the mind works (Newell, 1990). However, these models are not complete, and the process of developing them is not well understood because it is challenging to comprehend how the mind processes or completes a task (Ritter et al., 2020). Moreover, defining the knowledge that a model needs to perform and predict is a complex undertaking. Although cognitive architectures are designed and developed to assist in this model development process, there is still a lack of information and tools (Gonzalez et al., 2003; Laird, 2019; Ritter et al., 2019).

ACT-R (Anderson, 1990; Anderson & Lebiere, 1998), a well-known cognitive architecture, is a set of programmable information processing mechanisms. It encompasses cognition and interaction with the environment (Byrne, 2012; Newell, 1990; Ritter et al., 2019; Tehranchi & Ritter, 2017). As a theory of cognition used to build models, ACT-R can predict and explain human behavior.

ACT-R distinguishes between two types of knowledge (Bothell, 2017): *declarative* and *procedural*. Declarative knowledge represents what we know, while procedural knowledge represents what we exhibit. The ACT-R architecture consists of a set of modules, such as *goal* module, *declarative* module, and *procedural* module, that communicate through an interface called a *buffer*. The goal module holds the information the model needs to perform its current task. The declarative module stores chunks found in declarative knowledge, while the procedural module holds rules found in procedural knowledge, which are called productions.

Researchers have used ACT-R to evaluate different ranges of learning schedules and have demonstrated its capability to implement short and long decay (Tehranchi et al., 2021). Various studies have investigated the learning curve demonstrated by ACT-R (Rasmussen, 1987; VanLehn, 1996). Kim, Ritter, and Koubek (Kim et al., 2013) discussed and implemented a learning theory in a complex implementation and recommended using optimized learning for short and long decays. This work investigates how individuals learn and activate corresponding declarative memory while making abacus gestures.

Abacus Gesture Study

In this section, we briefly describe the procedure for collecting human-subject data in our prior work (Ehtesham-Ul-Haque & Billah, 2023), which we utilized to evaluate our cognitive models in the present study. The aim of the previous study was twofold: (1) to collect data on the time required to make different abacus gestures across multiple blocks; and (2) to evaluate an algorithm using an off-the-shelf computer vision library to detect abacus gestures from a 2D commodity camera. The study recruited 20 participants, including 19 right-handed individuals and 1 left-handed individual. The study was designed with three distinct phases.

The first phase was designed for practice. The participants were instructed on the worth of each finger and how to make different numbers (integers from 0 to 99) by opening and closing their fingers. They were also informed that some numbers can be made in multiple ways. For example, to make 26, one can open any two fingers (not the thumb) on their non-dominant hand and open the thumb and any one finger on the dominant hand. Next, they practiced making some abacus gestures for a few given numbers. Participants continued practicing until they became familiar with abacus gestures before moving forward to the next phase.

The second phase involved participants making a series of abacus gestures, one at a time, each corresponding to a non-zero random number (between 1 and 99) shown on the screen. Each task block contained 20 such gestures, and there was a total of 5 such blocks. The gesture representing the number zero (i.e., all fingers are closed) was a delimiter, reserved to mark the end of making a non-zero abacus gesture. As soon as the participants made a gesture and returned to the zero state, the study conductor pressed a button to show the next random number. Participants were allowed to take a rest between the blocks. In total, each participant completed 100 abacus gestures (= 5 blocks × 20 gestures/block). Each session was audio-video recorded, and we manually annotated the video to measure the gesture-making time.

In the third phase, similar to the second phase, participants were instructed to make a series of abacus gestures shown in a sentence containing 10 two-digit numbers in front of a commodity camera. Participants used the zero state to delimit between two gestures. This phase involved the computer vision module to accurately detect the abacus gestures. The response time was the duration between the computer generating a random number on the screen and detecting the participant's abacus gesture.

We used the observed human data generated in the second phase since our models aim to focus on human response times, excluding the processing time from the computer.

Modeling

We developed three cognitive models using ACT-R 7.26 within Emacs 29.1[1], an extensible and customizable editor. ACT-R was written in the Lisp programming language, and Emacs served as a compiling platform. The model output included a trace of modules, buffers, and fired productions, along with the predicted response time. The response time was the primary output for this study since we compared the average response times obtained from the three cognitive models. For the learning phase, we assume that the participants have already acquired the necessary knowledge and their declarative knowledge with different structures have been defined.

Model 1

In this model, we hypothesized that participants can express abacus gestures by quickly opening the required fingers all at once. We also hypothesized that participants could express each of the 100 abacus gestures more rapidly as they proceeded through task blocks.

A single chunk type with 10 slots was defined, each representing one of the 10 fingers. In the declarative module, we stored 100 chunks (integers from 0 to 99). In the production rules, the model received a random number from 0 to 99, searched for the corresponding chunk, and repeated these steps 100 times. Figure 2 shows how the number 26 was defined in Model 1.

```
(n26 ISA number number 26
    non-dominant-index   1
    non-dominant-middle  1
    non-dominant-ring    0
    non-dominant-little  0
    dominant-index  1
    dominant-middle 0
    dominant-ring   0
    dominant-little 0
    non-dominant-thumb   0
    dominant-thumb  1
     )
```

Figure 2: The abacus gesture for the number 26 in Model 1's chunk. To represent 26, the index and middle fingers of the non-dominant hand are raised (as indicated by the Boolean flag 1), while the index finger and thumb of the dominant hand are raised (also indicated by the Boolean flag 1). All other fingers are closed, as indicated by the Boolean flag 0.

Recall that for some numbers, there exist multiple possible finger combinations. However, Model 1 only considered one possible combination and defined only one chunk for each number. After defining declarative knowledge, the cognitive model defines production rules to simulate the cognitive process in the human mind for making abacus gestures. These production rules constitute the procedural knowledge of the model.

A function generates a random number, and the model places it in the retrieval buffer. The model reads the number in the production rule called "start". Then, the model retrieves the representation of the number by using the retrieval buffer to recall the specific chunk matching the information previously saved by the model. Model 1 recalls the chunk, such as the one shown in Figure 3, after the production rule "start" is fired. The selected chunk with the values in its slots is stored in the production rule variables. The last step is to output the representation of the number using the Boolean flags 0 and 1, as shown in Figures 3 and 4. The second production rule, called "search", completes these steps.

```
   102.796   DECLARATIVE            RETRIEVED-CHUNK N26
   102.796   DECLARATIVE            SET-BUFFER-CHUNK RETRIEVAL N26
   102.796   PROCEDURAL             CONFLICT-RESOLUTION
   102.846   PROCEDURAL             PRODUCTION-FIRED FIRST
26 VALUE
0 NON-DOMINANT-LITTLE
0 NON-DOMINANT-RING
1 NON-DOMINANT-MIDDLE
1 NON-DOMINANT-INDEX
0 NON-DOMINANT-THUMB
1 DOMINANT-THUMB
1 DOMINANT-INDEX
0 DOMINANT-MIDDLE
0 DOMINANT-RING
0 DOMINANT-LITTLE
```

Figure 3: Sample ACT-R model trace of thoughts for outputting the number 26. The first column in the initial four rows shows the running time in seconds.

The workflow of Model 1 is shown in the left side of Figure 4. It demonstrates a flowchart of the steps needed in each block containing 20 gestures for Model 1.

Model 2

Model 2 was developed based on feedback regarding user experience in our prior study, where participants mentioned that focusing on both hands simultaneously could be challenging at times (Ehtesham-Ul-Haque & Billah, 2023). For instance, the fifth participant said, "*When both of my hands are involved, this attention gets divided, and I need to make sure I am combining the correct fingers from both hands.*" She also stated, "*Although I need to focus on both hands when making numbers like 87, I can take my time and divide the attention. For example, I can focus on my left hand first to make an 80, and the system also shows 80. Then it waits until I focus on my right hand to make a 7*". This feedback suggests that participants are more likely to focus on their non-dominant hand to form tens and then focus on the dominant hand to form the unit digits.

[1] The code for the models can be found at https://github.com/HCAI-Lab/Abacus-Gestures-Code-for-CogSci-2024.

The defined chunks in Model 2 were similar to those in Model 1, but the difference was that it had 19 chunks to represent numbers from 0 to 9 and 10, 20, …, 90. There were three production rules in Model 2. The first production rule generated the number and read the numbers in tens and units separately. Then, Model 2 retrieved and reported the numbers in two different production rules. The second production rule outputted the representation of the non-dominant hand, while the third production rule outputted the representation of the dominant hand. The workflow of Model 2 is also shown in the right side of Figure 4.

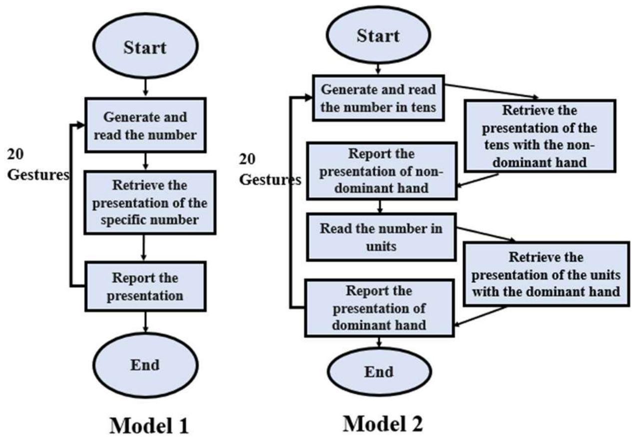

Figure 4: Work Flowchart of the Model 1 and Model 2.

Model 3

Model 3 was developed with only 14 chunks. It categorized the fingers into four chunk types based on non-dominant vs. dominant hands and thumb vs. non-thumb fingers. We used the Boolean flag 0 or 1 for the thumb to represent its closed or open state, respectively. For the other non-thumb fingers, the numbers 0, 1, 2, 3, and 4 were used, representing the total number of open fingers. This approach decreased the number of chunks required. There were five production rules: the first production rule started the process, while the other four production rules generated the numbers for the non-dominant thumb, non-dominant other fingers, dominant thumb, and dominant other fingers and outputted them.

For example, to represent the number 26, the following steps are taken: (i) The number of tens is less than 50, so the thumb of the non-dominant hand is closed. (ii) The number of tens is 20, so two fingers of the non-dominant hand (excluding the thumb) are open. (iii) For the unit digit, 6 is greater than 5, so the thumb of the dominant hand is open. (iv) The remaining value is 6-5=1, so one finger of the dominant hand (excluding the thumb) is open. The workflow of Model 3 is shown in Figure 5.

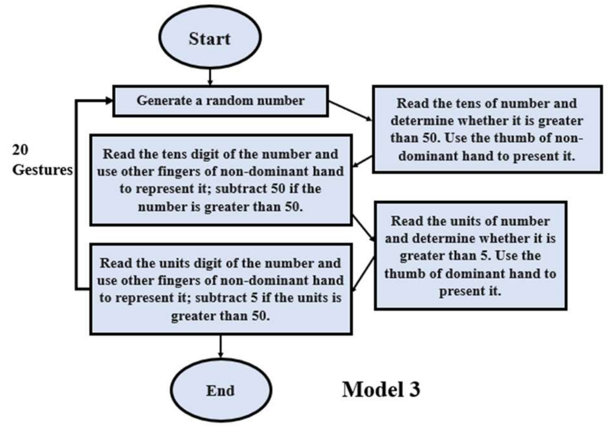

Figure 5: Work Flowchart of the Model 3.

Model Output & Parameters

We now describe the model parameters we scaled for the three cognitive models. The response time, the primary output for this work, consists mainly of the sum of each chunk's retrieval time – the time the declarative module takes to respond to each chunk (Bothell, 2017). In ACT-R, retrieval time includes various mathematical equations, displaying different outcomes based on their parameters. The retrieval time for a chunk i is given by this equation:

$$Time\ (s) = Fe^{-(f*A_i)}$$

which includes three parameters: activation value (A_i), latency factor (F, default = 1 second), and latency exponent (f, default = 1 second). Activation value particularly controls how long it takes for the chunk to be retrieved. The activation value is defined as

$$A_i = B_i + \epsilon_i$$

indicating the sum of base-level activation (B_i) and noise (ϵ_i). The base-level activation increases as the model uses specific chunks more recently and frequently. Base-level activation can be calculated differently based on the optimized learning (:ol) parameter. We adopted the optimized base-level activation to better predict the human mind. The optimized base-level activation of chunk *i* is calculated as

$$B_i = \ln(n/(1-d)) - d * \ln(L)$$

where n indicates the number of presentations of chunk i, L indicates the time since the chunk was created, and d indicates the decay parameter. Table 1 shows configurations for three cognitive models, which make them more consistent with human data.

Table 1: Configurations for three models (ol indicates whether optimized learning is enabled. 'T' means it is enabled. Other parameters are discussed in the Model Output & Parameters section.)

	DEFAULT VALUE	MODEL 1	MODEL 2	MODEL 3
d	0	0.1	0.01	0.01
ol	F	T	T	T
F	1	1	1.22	0.66
f	1	1	0.16	0.16
ϵ	0	0.5	0.5	0.5

Results

The models' responses and corresponding decelerative memory activation values are used to illustrate the models' learning curve in our abacus gesture study.

Comparison with Human Responses

Performance metrics: We chose the Root Mean Squared Error (RMSE) to compute the average magnitude of errors between the models' predicted responses and human responses in different blocks. This metric makes it easier to compare the performance of our three proposed models; the lower the value, the better. Besides RMSE, we also report the correlation between the responses of a model and human responses (the higher the value, the better).

Human responses: We utilized human responses for each block – 2.42, 2.14, 1.98, 1.87, and 1.84 seconds – as shown by the blue line in Figure 6. The human responses show a clear learning curve. On average, humans take 2.42 seconds to make an abacus gesture, which converges to 1.84 seconds over five blocks.

Model 1 responses: Similar to the human study, this model outputted the abacus representation of 20 random numbers in each block. The average response times for these blocks are 1.75, 1.88, 1.67, 1.77, and 1.81 seconds, as shown by the red line in Figure 6. On average, the response time for one abacus gesture is 1.78 seconds. Compared to human responses, these numbers do not indicate any learning. Even though there is a decrease in response time for block 3, with a sign that the difference with human data is less than 0.01 for block 5, the RMSE between Model 1's response and the human responses is still noticeable, as shown in Table 2.

This discrepancy strongly suggests that our hypothesis in Model 1, that participants might have remembered the knowledge required to make the corresponding abacus gesture for any random number, is not correct. It also suggests that (a) the usage of two hands simultaneously limits the learning behavior, and consideration of the dominant hand and order of using hands needs to be included; and (b) 100 chunks need to be broken into meaningful gestures to indicate learning.

Recall that in the human experiment, participants could make new abacus gestures by transferring knowledge from making previous gestures. For example, the numbers 86 and 87 share the same gestures for the non-dominant hand, meaning that participants could have reduced the time to make 87 if they already knew how to make 86. This likely caused the differences in response time and learning curve.

Model 2 and Model 3 responses: Model 2 and Model 3 overcome the limitations of Model 1 by considering the dominant hand and non-dominant hand separately and utilizing a smaller number of chunks – 19 chunks for Model 2 and 14 chunks for Model 3. As such, their learning process is closely correlated with human data (Figure 6). Table 2 shows both Model 2 and Model 3 have smaller RMSEs and higher correlations, compared to Model 1.

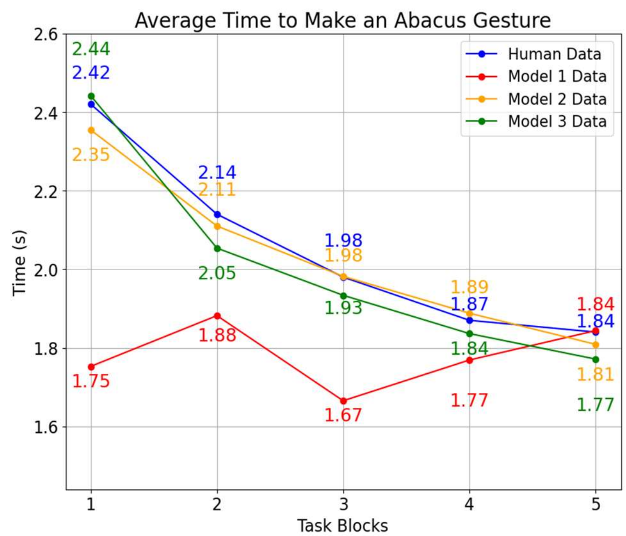

Figure 6: Average response time for five blocks by humans and the three proposed models.

Table 2: Comparing the fit of learning curves to human data for the Abacus Gestures.

Model	RMSE (Human vs. Model)	Correlation (Human vs. Model)
Model 1	0.352	- 0.053
Model 2	0.036	0.995
Model 3	0.056	0.992

Activation value: The activation value is obtained from 1 round consisting of 100 abacus gestures. For every 20 abacus gestures, the average activation values are calculated. The activation values for model 1 do not show a trend, meaning that model 1 does not learn from the process and the retrieval time did not change. Results show the improvement for models 2 and 3. In Figure 7, the activation values of model 2 and model 3 increase, suggesting that models are learning from the process.

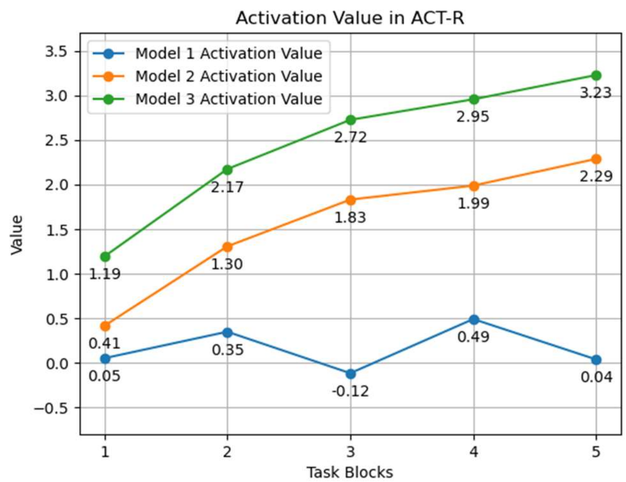

Figure 7: Activation value for five blocks (20 abacus gestures per block, totaling 100 abacus gestures).

Discussion and Future Work

We developed three models to predict the response time for representing abacus gestures to answer our research questions. *Can a cognitive model predict human mind behavior by using abacus gestures to represent the numbers 0-99?* The answer is yes -- all three models were able to complete the task and successfully represent abacus gestures. *How accurate are the predictions made by cognitive models?* While Model 1 does not perform very well, Model 2 and Model 3 demonstrate strong correlations with the human data (0.995 and 0.992, respectively), indicating human-like learning curves with increased activation values. *What factors contribute to the process of learning abacus gestures?* These models provide several insights, including a new type of input (i.e., mid-air gestures) in cognitive models, predicting response times for a repetitive task, and developing the learning curves for the ACT-R procedure. These will inform researchers to understand the cognitive processes in the human mind and how the knowledge of the abacus gestures can be represented, as well as retrieved to perform basic arithmetic operations mentally.

Limitations*:* One limitation of our current work is that it does not consider the time required for the physical movement of fingers while making abacus gestures. Our models assume that the time to generate a gesture is solely based on the cognitive processes involved in retrieving the correct representation from memory. However, in reality, there is an additional time component associated with the motor execution of the gesture, which can vary depending on factors such as the complexity of the gesture and the dexterity of the individual. Incorporating this motor execution time into the models would provide a more accurate representation of the total time required to generate abacus gestures.

Another limitation of this work is that it does not use the vision module. We plan to expand this work by adding a vision module and using interaction tools such as VisiTor (Bagherzadeh & Tehranchi, 2022) to simulate vision and motor processes more accurately.

Yet another limitation of our current models is the lack of error handling. In real-world scenarios, individuals may make errors while performing abacus gestures, such as incorrect finger positions or miscalculations. Our models do not account for these errors and their potential impact on learning and response times. We plan to incorporate error detection and correction mechanisms, as well as learning from errors (Bagherzadehkhorasani & Tehranchi, 2023; Lebiere, 1999) in order to provide a more realistic representation of the cognitive processes involved in abacus gesture learning.

Conclusion

This work enhances our understanding of how the human mind processes a comprehensive set of 100 numeric gestures that can be used in mid-air interaction. Our findings suggest that the human mind is more likely to represent these gestures by dividing attention between two hands rather than memorizing and outputting all gestures directly. Furthermore, we have demonstrated that using fewer chunks in cognitive models is closely correlated with human behavior and increased proficiency, as segmenting chunks can increase both the frequency and recency of information retrieval. These insights have important implications for future research in cognitive science and human-computer interaction. Adding a motor module represents an opportunity to expand our work and cognitive architectures. Furthermore, developing new heuristics that reduce the potential for overfitting of model parameters is another opportunity to expand our work.

Acknowledgments

We appreciate anonymous reviewers for their insightful reviews and comments. This work was supported by The Pennsylvania State University. We thank Human-Centered AI and A11y Lab members at PSU for their help.

References

Anderson, J. (1990). The adaptive character of thought. Hillsdale, NJ: Lawrence ErlbaumAssociates. *Inc. AndersonThe adaptive character of thought1990*.

Anderson, J., & Lebiere, C. (1998). Hybrid modeling of cognition: Review of the atomic components of thought. In: Erlbaum.[AGB].

Anderson, J. R., Matessa, M., & Lebiere, C. (1997). ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*, *12*(4), 439-460.

Bagherzadeh, A., & Tehranchi, F. (2022). Comparing cognitive, cognitive instance-based, and reinforcement learning models in an interactive task. Proceedings of ICCM-2022-20th International Conference on Cognitive Modeling, Bagherzadehkhorasani, A., & Tehranchi, F. (2023). Automatic Error Model (AEM) for User Interface Design: A new approach to Include Errors and Error Corrections in a Cognitive User Model. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *67*(1), 644-649. https://doi.org/10.1177/21695067231192248

Bothell, D. (2017). ACT-R 7 Reference Manual. In. act-r.psy.cmu.edu/wordpress/wp-content/themes/ACT-R/actr7/reference-manual.pdf.

Byrne, M. D. (2012). Unified theories of cognition. *Wiley interdisciplinary reviews. Cognitive science*, *3 4*, 431-438.

Cho, P. S., & So, W. C. (2018). A Feel for Numbers: The Changing Role of Gesture in Manipulating the Mental Representation of an Abacus Among Children at Different Skill Levels. *Front Psychol*, *9*, 1267. https://doi.org/10.3389/fpsyg.2018.01267

Ehtesham-Ul-Haque, M., & Billah, S. M. (2023). Abacus Gestures: A Large Set of Math-Based Usable Finger-Counting Gestures for Mid-Air Interactions. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, *7*(3), Article 93. https://doi.org/10.1145/3610898

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, *27*(4), 591-635.

Kim, J. W., Ritter, F. E., & Koubek, R. J. (2013). An integrated theory for improved skill acquisition and retention in the three stages of learning. *Theoretical Issues in Ergonomics Science*, *14*(1), 22-37.

Laird, J. E. (2019). *The Soar cognitive architecture*. MIT press.

Lebiere, C. (1999). The dynamics of cognition: An ACT-R model of cognitive arithmetic. *Kognitionswissenschaft*, *8*, 5-19. https://doi.org/10.1007/s001970050071

Newell, A. (1990). *Unified theories of cognition*. Harvard University Press.

Pai, H. Y., Pai, S. J., & Leonard, J. (1981). *The Complete Book of Chisanbop: Original Finger Calculation Method Created by Sung Jin Pai and Hang Young Pai*. Van Nostrand Reinhold Company.

Rasmussen, J. (1987). *Information processing and human-machine interaction. An approach to cognitive engineering*. North-Holland.

Ritter, F. E., Tehranchi, F., Dancy, C. L., & Kase, S. E. (2020). Some futures for cognitive modeling and architectures: design patterns for including better interaction with the world, moderators, and improved model to data fits (and so can you). *Computational and Mathematical Organization Theory*, 1-29.

Ritter, F. E., Tehranchi, F., & Oury, J. D. (2018). ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, e1488.

Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *Wiley Interdiscip Rev Cogn Sci*, *10*(3), e1488. https://doi.org/10.1002/wcs.1488

Tehranchi, F., Oury, J. D., & Ritter, F. E. (2021). Predicting learning and retention of a complex task using a cognitive architecture. Proceedings of the Annual Meeting of the Cognitive Science Society,

Tehranchi, F., & Ritter, F. E. (2017). An eyes and hands model for cognitive architectures to interact with user interfaces. MAICS, The 28th Modern Artificial Intelligence and Cognitive Science Conference, Fort Wayne, IN: Purdue University.

VanLehn, K. (1996). Cognitive skill acquisition. *Annual review of psychology*, *47*(1), 513-539.