Analysis of a Common Neural Component for Finger Gnosis and Magnitude Comparison

Terrence C. Stewart (tcstewar@uwaterloo.ca)
Centre for Theoretical Neuroscience, University of Waterloo,
200 University Avenue W, Waterloo, ON, N2L 3G1, Canada

Marcie Penner-Wilger (mpennerw@uwo.ca)
Department of Psychology, King's University College at Western University
266 Epworth Avenue, London, ON, Canada N6A 2M3

Abstract

We recently developed a spiking neuron model that performs magnitude comparison and finger gnosis tasks using a common underlying neural system, explaining why performance on these tasks is associated in humans. Here, we explore the parameters in the model that may vary across individuals, generating predictions of error patterns across the two tasks. Furthermore, we also examine the neural representation of numbers in the magnitude comparison task. Surprisingly, we find that the model fits human performance only when the neural representations for each number are *not* related to each other. That is, the representation for TWO is no more similar to THREE than it is to NINE.

Keywords: magnitude comparison; finger gnosis; neural engineering; number representation

Introduction

We have recently proposed a neural model of a cognitive component underlying two disparate tasks: finger gnosis and magnitude comparison (Stewart et al., 2017). These tasks have been shown to be related via behavioural, fMRI imaging, and stimulation experiments, and our model describes a neural system that could be involved in both tasks, explaining this relation. However, in the initial paper, we did not perform an analysis of the effects of parameter variation on this model. Our goal in this paper is to present this parameter analysis in order to better understand the performance of this model.

Finger gnosis is the ability to indicate which fingers have been touched, out of the view of the participant. Typically two fingers (on the same hand) are touched while the participant's hand is occluded, and they must then indicate which fingers were touched (Baron, 2004).

The *magnitude comparison* task considered here is symbolic single-digit number comparison. Participants are visually shown two single-digit numbers and they are asked to indicate which one is larger.

Individual performance on the finger gnosis task predicts a variety of mathematical measures in both children (Fayol et al., 1998; Noel, 2005; Penner-Wilger et al., 2007, 2009) and in adults (Penner-Wilger et al., 2014, 2015). In particular, this relation is partially mediated by performance on the single-digit symbolic magnitude comparison task used here (Penner-Wilger el al., 2009, in prep.). Individuals who perform better at magnitude comparison also perform better at the finger gnosis task.

In addition to this behavioural result, representation of number and finger gnosis both activate the same brain regions (Andres, Michaux & Pesenti, 2012; Dehaene et al., 1996; Zago et al., 2001), both tasks are disrupted by rTMS and direct cortical stimulation to the same regions (Rusconi, Walsh, & Butterworth, 2005; Roux et al., 2003), and the tasks interfere with each other when performed at the same time (Brozzoli et al., 2008). For these reasons, we believe that there is a common component underlying these tasks. In other words, there is some set of neurons performing some operation that is used in each task. This makes it an example of neural *redeployment* (Penner-Wilger & Anderson, 2008, 2013)

In the current paper, we first outline in more detail a model that performs finger gnosis and number comparison, which we initially reported in Stewart et al. (2017). Second, we examine the behavioural effects in these two tasks, as different parameters in the neural model are varied. Given that the same neural components are used, changing an aspect of the model will affect both tasks. The results of these variations form a set of predictions about individual differences in performance on these tasks. Finally, we examine a parameter that only exists for the magnitude comparison task. Here, we need to decide how the different numbers are represented neurally. One possibility is to assume that the number SEVEN should have a neural representation that is more similar to the neural representation for EIGHT than it is to ONE, as this might explain why more mistakes are made when the numbers being compared are close to each other. As is shown below, the modelling result instead shows that there is a better match to human performance if the neural representation for each number is unrelated to the others.

A Common Component

We postulate that the shared system for these two tasks is a neural implementation of *an array of pointers*. That is, a neural system that can store a small set of arbitrary values, each of which can represent something. For example, one pointer could be set to the neural representation of the number SEVEN, while another pointer could be set to represent concepts like DOG or CAT or BLUE or QUIET or TOUCHED.

Importantly, we do not need to make a strong claim about the nature of the neural representation of these concepts here. Instead, we merely make the weak claim that there is some pattern of neural activity for each concept, and all we need is a system that can store an arbitrary pattern. Here we generally randomly choose these patterns, but we investigate what happens when these patterns are related to each other below.

To describe this system mathematically, we can say that there are a small set of vectors p_1, p_2, p_3, p_4, and p_5, one for each pointer. In order for this system to maintain information over time, if there is no input, these values should stay as they are. However, if there is an input, we also need to indicate which pointer(s) will be changed. To do this, we introduce a mask m, which controls which pointers will be affected by the input x. For example, if m=[0,1,0,0,0], then the second pointer p_2 will be affected by the input x. Mathematically, this can be written as:

$$p_i \Leftarrow \begin{cases} p_i & \text{if } m_i = 0 \\ x & \text{if } m_i = 1 \end{cases} \quad \text{(Eq. 1)}$$

For the magnitude comparison task, this would be used as follows. First, the vector for one of the numbers (e.g. SEVEN) would be loaded into the first pointer value by setting x to the vector for SEVEN and setting m to [1,0,0,0,0]. Next the other number (e.g. THREE) would be loaded into the second pointer by setting x to THREE and m to [0,1,0,0,0]. Over time, the values stored in the pointers would be as follows:

Magnitude Comparison Task

step	*x*	*m*	p_1	p_2	p_3	p_4	p_5
1	--	00000	--	--	--	--	--
2	SEVEN	10000	SEVEN	--	--	--	--
3	THREE	01000	SEVEN	THREE	--	--	--
4	--	00000	SEVEN	THREE	--	--	--

Once these values are stored in the pointers, the rest of the task can be completed by reading the values out and performing the comparison. The details for this are provided below.

For the finger gnosis task, a similar process is followed, but we use the pointers in a different way. In particular, the value that is being loaded in is always the vector for TOUCHED (indicating that this finger was touched), but the particular pointer that we load it into is what is important. In the following chart, we show the process when the second and fourth fingers are touched.

Finger Gnosis Task

step	*x*	*m*	p_1	p_2	p_3	p_4	p_5
1	--	00000	--	--	--	--	--
2	TOUCHED	01000	--	TOUCHED	--	--	--
3	TOUCHED	00010	--	TOUCHED	--	TOUCHED	--
4	--	00000	--	TOUCHED	--	TOUCHED	--

Once these values are loaded in, the rest of the finger gnosis task involves reading out these values and reporting them, as is detailed below. Importantly, while the remainder of the finger gnosis task is quite different from the magnitude comparison task, both tasks make use of this same array of pointers component.

Neural Implementation

Though this basic idea of an array of pointers is simple (and, indeed, is trivial to implement in a traditional computational model), here we implement this system using spiking neurons. The important point here is that neurons will not perfectly implement this algorithm; rather, their actual behaviour will only approximate this ideal. Importantly, this approximation can serve as an explanation for the mistakes made by people performing these tasks. Furthermore, changing the details of this neural implementation (for example, how many neurons are used, or how strong the mask is) can change the resulting behaviour, providing an explanation for the individual differences, and how errors on one task relate to errors on the other task.

To convert this model to spiking neurons, we use the Neural Engineering Framework (Eliasmith & Anderson, 2003). In this approach, different groups of neurons are used to represent each vector (e.g. x or p_1). Connections between groups of neurons implement functions on those variables. For example, if one group of neurons represents x and another group of neurons represents y, then we can form a connection from x to y such that $y=f(x)$. Given any particular function f, we can solve for the optimal synaptic connection weights between those groups of neurons that will best approximate that function.

When we solve for these synaptic connection weights, we are not making any claim about how these connection weights are learned, or how they are formed in a developmental process. Rather, we are simply finding the best possible way that the given neurons can perform this task, and leaving these larger developmental questions to future research.

With this in mind, our model is presented in Figure 1. Each box represents a group of neurons representing one vector. Arrows between boxes indicate connections between groups of neurons. In each case, these connections are optimized to compute the *identity function*. This is the simple function that just transmits information without changing it in any way.

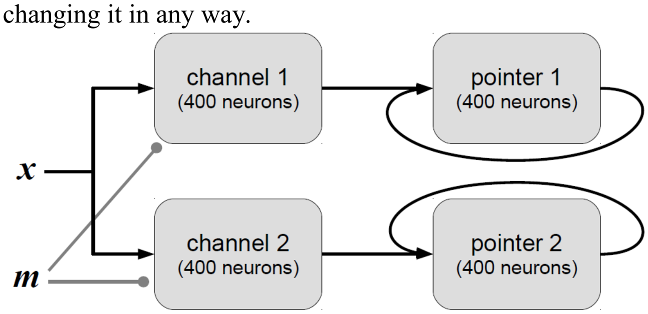

Figure 1: A neural implementation of an array of pointers. Only two pointers are shown.

The recurrent connection on the pointer neurons indicates that those 400 neurons are recurrently connected to themselves such that they will pass their own information back to themselves. In other words, whatever pattern of neural activity is generated in that group *will be self-sustaining*. That is, the pattern of activity will be maintained over time. Of course, since neurons are not perfect, this pattern will not be perfectly maintained, leading to a gradual decay of this memory system.

To load values into this system, we place the desired vector as the input x. This will drive the various *channel* neural populations to fire, representing that value x. This will then in turn drive the pointer populations to store that value. However, with just this system, any x value as input would be loaded into *all* of the pointers. In order to implement Equation 1 completely, we need a *mask* term to control which pointers will be affected. We accomplish this by selectively inhibiting the activity of the *channel* populations. If a channel is inhibited, the corresponding pointer population will not be affected by x.

As described more completely in Stewart et al. (2017), we implement all of this using standard Leaky Integrate-and-Fire spiking neurons using the simulation software Nengo (Bekolay et al, 2014). The resulting behaviour of the system loading two pointers (FIVE and SEVEN) is shown in Figure 2.

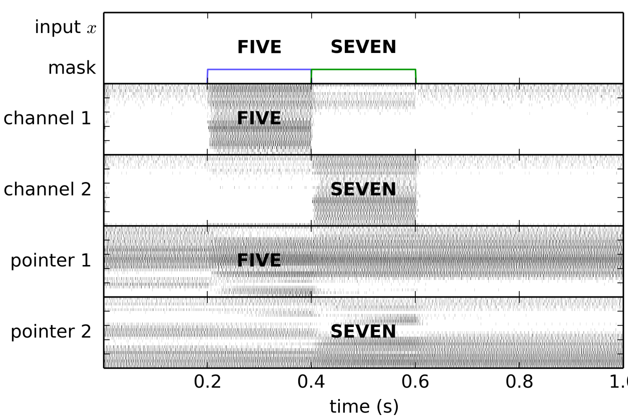

Figure 2: Spiking activity for an example magnitude comparison task. Top row shows input to the model. Other rows show spiking neuron activity over time. The text indicates which vector $\boldsymbol{x}$ is represented by the pattern of activity. Note that pointer 1 and pointer 2 maintain their spiking pattern (approximately) after the input has been removed. Figure from (Stewart et al., 2017).

In order to perform the two separate tasks, we then connect this same common component to one of two different output systems. For the finger gnosis task, the output is simply the identity function again, as all we need to do is to report the information stored in the pointers.

For the magnitude comparison task, we need a slightly more complex output. Rather than reporting the two stored numbers, we need to report whether the first number is larger or smaller than the second number. This is, itself, a function. So, in order to compute this, we use the NEF to solve for the optimal connection weights that will best approximate the function that maps the vectors for the two numbers to a single scalar output that is +1 if the first number is larger, and -1 if the second number is larger. We can think of this as training a group of neurons to memorize this list of desired inputs and outputs:

input	**output**
[ONE, TWO]	-1
[TWO, ONE]	+1
[ONE, THREE]	-1
[THREE, ONE]	+1
[TWO, THREE]	-1
...	...
[NINE, EIGHT]	+1

When we run this model, we treat a positive output as selecting the first number, and a negative output as selecting the second number. In Stewart et al. (2017) we also use the magnitude of this output to predict reaction times, but do not do that here.

Results

The basic results presented in Stewart et al. (2017) are shown in Figure 3. This includes both the model result and the empirical result gathered from human participants. We plot the percent error for the magnitude comparison task and for the finger gnosis task. Importantly, we fit the model parameters based on the magnitude comparison task *only*, leaving the finger gnosis task as a pure prediction based on those same parameter values.

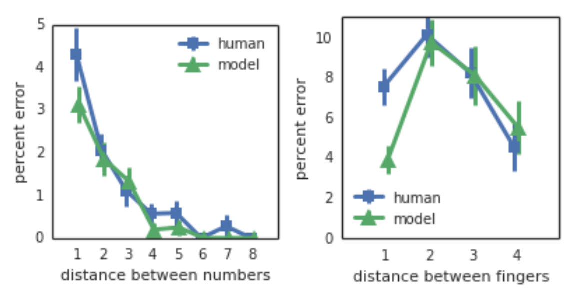

Figure 3: Best-fit model results for the magnitude comparison task (left) and finger gnosis (right). Parameters are fit on the magnitude task and then applied to the finger gnosis task. Standard errors are shown.

The best-fit parameters are as follows:

parameter		**value**
# neurons for combining pointer values:	n_c	1000
Standard deviation of training noise:	$noise_t$	0.15
Amount of channel inhibition:	c	0.875
Dimensionality of x vector:	D	8
Uniqueness of digit representation:	u	1.0

This core result indicates that the model captures the basic characteristics of the behavioural data. For magnitude comparison, we see the standard *distance effect*, where numbers that are farther apart (e.g. 2 and 7) are easier than numbers that are closer together (e.g. 5 and 6). We see a similar effect for finger gnosis, with the exception of when two fingers right next to each other are touched. In the human participant data, fingers next to each other are easier than fingers that are two apart. The model shows this same effect, but it is much more pronounced.

Parameter Exploration

To further characterize this model, we systematically varied these parameters. Importantly, since our theory is that both of these tasks use the same common neural system, whatever parameter value is used for one task should also be used for the other task.

However, this is only true for an individual person. It is plausible that, if this model is correct, different people may have different parameter values for this system. Thus, by changing these parameter values we make predictions about how performance on these two tasks may co-vary in individuals.

Parameter 1: n_c

The first parameter is the number of neurons to use to combine together the outputs from all of the pointers. Once combined together in this way, we can either create output connections that compute which of the two numbers is biggest (for magnitude comparison) or that just compute the identity function (for finger gnosis). However, the accuracy of this computation will be affected by the number of neurons used. This is shown in Figure 4.

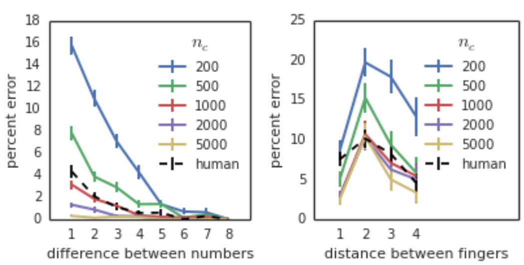

Figure 4: The effects of varying the number of neurons used to combine the represented pointer values together for the magnitude comparison task (left) and the finger gnosis task (right). Standard errors are shown.

From this, we note that 500 or fewer neurons gives significantly higher error rates than the mean human performance on both tasks. Having more than 1000 neurons gives improved performance for the magnitude comparison task and most of the finger gnosis task, but does not improve the peak error at a finger distance of two.

Parameter 2: $noise_t$

Next, we look a the amount of random noise used when finding the connection weights out of this combined population. That is, the neural activity from this combined population must cause change in a separate population that represents the network's response to the task. This change is, of course, due to synaptic connection weights. When we use the Neural Engineering Framework to solve for these weights, we can specify how much random variability is added. The right amount noise should make the network more robust to random variations, but too much noise will cause it to lose accuracy. The results are shown in Figure 5.

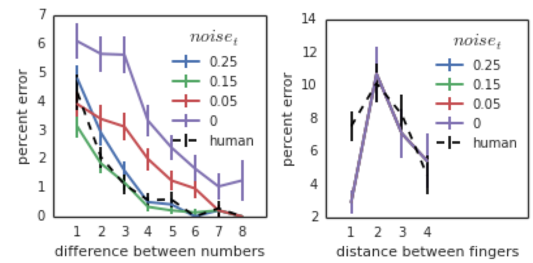

Figure 5: The effects of varying the amount of noise used in training the weights for the tasks for the magnitude comparison task (left) and the finger gnosis task (right). Standard errors are shown.

For the magnitude comparison task, we see the expected effect where there is an optimal value for this noise (0.15). Less noise than this gives extremely poor results for all distances. Interestingly, having more noise than this only increases the error for small differences between the numbers.

For the finger gnosis task, we get the surprising result that the model is unaffected by the amount of noise.

Parameter 3: c

Next, we examine the inhibition factor which turns off the channels leading into each pointer. With a value of 1, this inhibition would perfectly inhibit all of the neurons in the non-active channels, leading to no activity in those channels, and thus no change in the other pointer values. If this is less than one, however, the neurons will not be perfectly “turned off”, and so there will be some small influence on the other pointers when one of them is set. For example, in Figure 2, we see some neural activity in the channels that are not being set, reflecting $c<1$. We assume this amount of inhibition scales linearly with the distance from the target item, so values sent into pointer 1 have more influence on pointer 2 than they do on pointer 3. Results are in Figure 6.

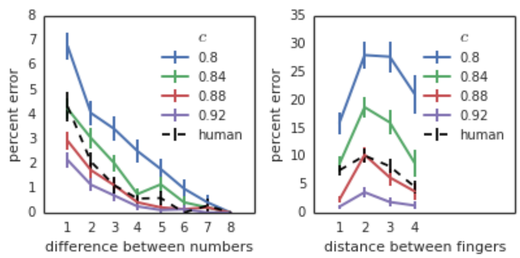

Figure 6: The effects of varying the amount of channel inhibition for the magnitude comparison task (left) and the finger gnosis task (right). Standard errors are shown.

In this case, we get a clear result that there is an optimal value for c (we found 0.875 best). Importantly, this optimal value works for both the magnitude comparison task and the finger gnosis task.

Parameter 4: *D*

Finally, we vary the dimensionality of the input stimulus x. This controls the degrees of freedom in the randomly chosen patterns for each represented concept (ONE, TWO, TOUCHED, etc). Results are shown in Figure 7.

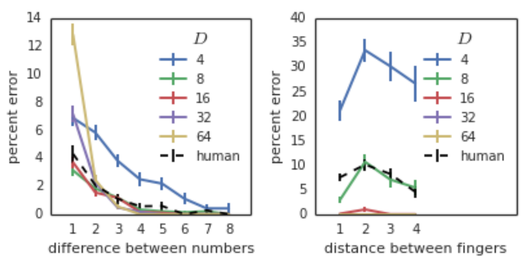

Figure 7: The effects of varying the dimensionality of *x* for the magnitude comparison task (left) and the finger gnosis task (right). Standard errors are shown.

Here, we see that the magnitude comparison task indicates that if *D* is too large (i.e. 32 or 64), it produces a large increase in the error, but just for the case where the difference between the numbers is 1. It also produces a large increase in error overall if *D* is too small (i.e. 4). For the finger gnosis task, small *D* produces a very large increase in error as well, but large *D* (above 8) causes a massive decrease in the error.

Number Representation

If we consider just the magnitude comparison task, there is a further parameter that is worth investigating. This is the question of how numbers are represented in the model. In particular, should the representation for TWO be more similar to THREE than it is to NINE? After all, as can be seen in the human data, participants are more likely to make mistakes when number are close to each other, which seems to imply that the neural activity for TWO should be more similar to the activity for THREE than it is to NINE.

As we are using a vector representation in this model, this becomes the question of how to choose what vector to use for ONE, TWO, THREE, etc. In the simplest case, we can choose these vectors completely randomly, so that there is no similarity structure. At the other extreme, we could randomly choose a vector for ONE, a different vector for NINE, and then smoothly interpolate between these two to create the vectors for TWO, THREE, FOUR, etc. To explore this, we define a parameter u which interpolates between fully random representation where each number is represented with a different random unique number (*u*=1.0) and fully structured representation where TWO is halfway between ONE and THREE (*u*=0.0).

The effects of varying this uniqueness parameter are shown in Figure 8. Crucially, if there is low uniqueness (i.e. if the neural representation of TWO is more similar to ONE than it is to NINE), then we reach a much higher error rate than is observed in the human data.

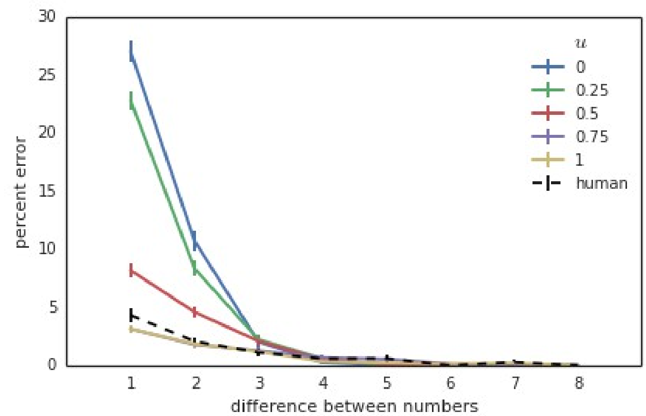

Figure 8: The effects of varying the uniqueness of the number representation in the magnitude comparison task. Standard errors are shown.

This was a surprising result for us. The observed error pattern in the human data (where numbers that are close to each other are more likely to produce errors) is *not* the result of the actual neural representation of the numbers being similar to each other. Rather, this pattern of errors is due to mistakes made in *extracting* the information from the group of neurons. When neurons are used to approximate the "which number is larger" function, the optimal connection weights lead to a system which is more likely to make mistakes between nearby numbers, *even though they are not "nearby" in terms of neural activity*. They are, however, nearby in terms of the function being computed.

Conclusions

We have examined the behaviour of a model of how magnitude comparison and finger gnosis can both rely on the same common neural component: a system for storing an array of pointers. Since this neural system is believed to be used in both tasks, by varying the parameters of this system we produced predictions of how an individual's error performance on both tasks can be related. However, it should be noted that all of the comparisons performed in this paper were to the *mean* human performance. The next step is to look at individual differences in this task and determine if the same patterns occur in the participant data. If it does, then we may have an explanation for this variation in terms of different people having different parameter settings for this common component.

Furthermore, we have a novel explanation as to why the *distance effect* exists. In our model, the distance effect (the fact that more errors are made when two numbers are close in magnitude) is *not* due to those two numbers having similar neural representations. Rather, the neural representation of each number is completely random. If we do impose some similarity in the neural representation, then the distance effect becomes much larger than it is in the participant data. This means that in our model, the distance effect emerges purely from the difficulties involved in generating synaptic connections that determine which of the two numbers is larger, rather than the more typical interpretation that it comes from similarities in the neural representation itself.

References

Andres, M., Michaux, N., & Pesenti, M. (2012). Common substrate for mental arithmetic and finger representation in the parietal cortex. *Neuroimage*, 62(3), 1520-1528.

Baron, I. S. (2004). *Neuropsychological Evaluation of the Child*. New York, NY: Oxford University Press.

Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T.C., Rasmussen, D., Choo, X., Voelker, A., & Eliasmith, C.. (2014). Nengo: a python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*

Brozzoli, C., Ishihara, M., Göbel, S. M., Salemme, R., Rossetti, Y., and Farnè, A. (2008). Touch perception reveals the dominance of spatial over digital representation of numbers. *Proc. Natl. Acad. Sci. U.S.A*. 105, 5644–5648.

Dehaene, S., Tzourio, N., Frak, V., Raynaud, L., Cohen, L., Mehler, J., & Mazoyer, B. (1996). Cerebral activations during number multiplication and comparison: A PET study. *Neuropsychologia,* 34, 1097-1106.

Eliasmith, C. & Anderson, C.. (2003). *Neural engineering.* MIT Press, Cambridge, MA.

Fayol, M., Barrouillet, P., & Marinthe, C. (1998). Predicting arithmetical achievement from neuro-psychological performance. *Cognition*, 68, B63-B70.

Noël, M.-P. (2005). Finger gnosia: A predictor of numerical abilities in children? *Child Neuropsychology*, 11, 413-430.

Rusconi, E., Walsh, V., & Butterworth, B. (2005). Dexterity with numbers: rTMS over left angular gyrus disrupts finger gnosis and number processing. *Neuropsychologia*, 43(11), 1609-1624.

Penner-Wilger, M., & Anderson, M. L. (2013). The relation between finger gnosis and mathematical ability: Why redeployment of neural circuits best explains the finding. *Frontiers in Psychology,* 4, 877.

Penner-Wilger, M., & Anderson, M. L. (2008). An alternative view of the relation between finger gnosis and math ability. *30th Cognitive Science Society*. 1647-1652.

Penner-Wilger, M., Fast, L., LeFevre, J., Smith-Chant, B. L., Skwarchuk, S., Kamawar, D., & Bisanz, J. (2007). The foundations of numeracy: Subitizing, finger gnosia, and fine-motor ability. *29th Cogn. Science Society,* 1385-1390

Penner-Wilger, M., Fast, L., LeFevre, J., Smith-Chant, B. L., Skwarchuk, S., Kamawar, D., & Bisanz, J. (2009). Subitizing, finger gnosis, and the representation of number. *31st Annual Cognitive Science Society*, 520-525

Penner-Wilger, M., Waring, R. J., & Newton, A. T. (2014). Subitizing and finger gnosis predict calculation fluency in adults. *36th Cognitive Science Society*, 1150-1155.

Penner-Wilger, M., Waring, R. J., Newton, A. T., & White, C. (2015). Finger gnosis and symbolic number comparison as robust predictors of adult numeracy. 37th Conference of the Cognitive Science Society, 2963.

Penner-Wilger, M., Waring, R. J., Newton, A. T., & White, C. (in prep). Finger gnosis as a robust predictor of numeracy in children and adults.

Roux, F.-E., Boetto, S., Sacko, O., Chollet, F., & Tremoulet, M. (2003). Writing, calculating, and finger recognition in the region of the angular gyrus: a cortical study of Gerstmann syndrome. *J. Neurosurgery*, 99, 716-727.

Stewart, T.C., Penner-Wilger, M., Waring, R., and Anderson, M. (2017). A Common Neural Component for Finger Gnosis and Magnitude Comparison. *Annual Meeting of the Cognitive Science Society.*

Zago, L., Pesenti, M., Mellet, E., Crivello, F., Mazoyer, B., & Tzourio-Mazoyer, N. (2001). Neural correlates of simple and complex mental calculation. *NeuroImage*, 13, 314-327.