A Comprehensive Model of Development on the Balance-scale Task

Fredéric Dandurand (frederic.dandurand@gmail.com)
Department of Psychology, Université de Montréal,
90 ave. Vincent-d'Indy, Montréal, QC H2V 2S9 Canada

Thomas R. Shultz (thomas.shultz@mcgill.ca)
Department of Psychology and School of Computer Science, McGill University,
1205 Penfield Avenue, Montreal, QC H3A 1B1 Canada
Tel: ++1 514 398 6139
Fax: ++1 514 398-4896

Abstract

We present a new model of children's performance on the balance-scale task, one of the most common benchmarks for computational modeling of psychological development. The model is based on two processing modules, called the intuitive and torque-rule modules, both implemented as constructive neural networks. While the intuitive module recruits non-linear sigmoid units as it learns to solve the task, the second module can additionally recruit a neurally-implemented torque rule, mimicking the explicit teaching of torque in secondary-school science classrooms. A third, selection module decides whether the intuitive module is likely to yield a correct response or whether the torque-rule module should be invoked on a given balance-scale problem. The model progresses through all four stages seen in children, ending with a genuine torque rule that can solve untrained problems that are only solvable by comparing torques. The model also simulates the torque-difference effect and the pattern of human response times, faster on simple problems than on conflict problems. The torque rule is more likely to be invoked on conflict problems than on simple problems and its emergence requires both explicit teaching and practice. Appendices report evidence that constructive neural networks can also acquire a genuine torque rule from examples alone and show that Latent Class Analysis discovers small, unreliable rule classes in both children and computational models.

Keywords: Cognitive development; balance scale; constructive neural networks; knowledge-based learning; KBCC; SDCC.

1. Introduction

Ongoing debates between symbolic and neural-network models of cognition have often focused on development of children's performance on balance-scale problems, one of the most simulated tasks in developmental psychology. The symbolic view is that knowledge is represented in propositional rules referring to things in the world, that processing occurs as rules are selected

and fired, and that knowledge is acquired by learning such rules. In neural-network accounts, active knowledge is represented in rapidly changing neuronal-unit activations and long-term knowledge by excitatory and inhibitory synaptic connections between units, processing involves activation passing from one layer of units to another, and knowledge acquisition results from adjustment of connection weights and perhaps recruitment of new units into the network. The symbolic approach has been referred to as rule *use*, and the neural-network approach as rule *following* (Shultz & Takane, 2007).

Although this may seem to be a subtle distinction, there are important differences between the two viewpoints that have consistently guided research over the last few decades. The rule-use approach assumes that people have and use rules to guide their reasoning and behavior, perhaps affording the perfect generalization that symbolic rules may allow. Rule-use is quite consistent with the idea that human cognition is often quite regular. In contrast, the rule-following approach assumes that such regularities may be naturally approximated by neural networks that adapt to regularities in the environment. This affords more graded generalizations whose regularity approximates the extent to which the environment is consistently regular, with the possible advantage that both regularities and exceptions can be accommodated within the same neural network. In rule-use systems, exceptions are instead typically memorized, and represented separately from the rules themselves. Such differences are highlighted in precise computational models of psychological theories (Shultz, 2003).

One of the most frequently modeled domains in developmental psychology focuses on the balance-scale task, studied by Siegler (1976) and others. The balance-scale is interesting because it is representative of the many tasks requiring integration of information across two separate quantitative dimensions and because it provides well-replicated results with an interesting stage progression.

Here we present a new computational model of balance-scale acquisition that addresses a recent criticism affecting many of the balance-scale computational models – ensuring that the final stage consists of a genuine, multiplicative torque rule and not a simpler rule based on addition (Quinlan, van der Maas, Jansen, Booij, & Rendell, 2007). After describing the balance-scale task and phenomena, we present our new computational model.

1.1 Balance-scale task and phenomena

The task presents several pegs positioned on a rigid beam at regular distances to the left and right of a fulcrum (Siegler, 1976). An experimenter places some identical weights on a peg on the left side and some number of identical weights on a peg on the right side of the beam. The participant is asked to predict which side of the scale will drop, or whether the scale will remain balanced, when the beam is released from its supports, usually a block placed under each end of the beam. Archimedes' principle of the lever describes a rule that yields a correct answer to all such problems: multiply the weight and distance from the fulcrum on each side and predict that the side with the larger product (or torque) to drop.

A neural-network simulation using the cascade-correlation (CC) algorithm (Shultz, Mareschal, & Schmidt, 1994) captured the four stages seen in children (Siegler, 1976): 1) predicting the side with more weights to descend, 2) when the weights are equal on both sides, also predicting the side with greater distance to descend, 3) predicting correctly when weight and distance cues both forecast the same result and performing at chance when these cues conflict, and 4) being correct on at least 80% of balance-scale problems.

1.1.1 Diagnosing stage 4

If performance at Stage 4 is diagnosed as being correct on 80% of balance-scale problems, some of which are difficult problems in which weight and distance cues conflict with each other, then at least some computational models, both symbolic (Schmidt & Ling, 1996) and connectionist cascade-correlation networks (Shultz, Mareschal, et al., 1994) reach Stage 4. But if Stage 4 is defined by possession of a genuine multiplicative torque rule, as opposed to say an addition rule, the modeling challenge remains open. Because many conflict problems can be solved by just adding weight and distance, documentation of a torque rule must be supported by success on problems that cannot alternately be solved by an addition rule (Boom, Hoijtink, & Kunnen, 2001; Quinlan, et al., 2007).

With five pegs and five weights, the problem size often used in simulations of the balance scale (Shultz, Mareschal, et al., 1994), there are 625 total problems, of which only 200 are relatively difficult conflict problems in which weight and distance information, used alone, predict different outcomes. Only 52 of these conflict problems are torque problems that cannot be solved by mere addition; the other 148 are addition problems that can be solved by adding distance and weight on each side and comparing these sums.

Addition was routinely ignored in computational models of balance-scale development, whether symbolic (Schmidt & Ling, 1996) or connectionist (McClelland, 1989; Schapiro & McClelland, 2009; Shultz, Mareschal, et al., 1994), just as it had been ignored in many older psychology experiments on the balance scale. But with evidence that at least some people use or follow a genuine torque rule, solving balance-scale problems that addition cannot solve (Boom, et al., 2001; Quinlan, et al., 2007), it becomes important to test computational models for their ability to acquire and use a genuine torque rule.

This problem of accurately diagnosing a terminal stage does not arise in the many other developmental domains where constructive neural networks have been successfully applied: conservation (Shultz, 1998, 2006), seriation (Mareschal & Shultz, 1999), transitivity (Shultz & Vogel, 2004), integration of cues for moving objects (Buckingham & Shultz, 2000), shift learning (Sirois & Shultz, 1998), deictic pronouns (Oshima-Takane, Takane, & Shultz, 1999; Shultz, Buckingham, & Oshima-Takane, 1994), word stress (Shultz & Gerken, 2005), syllable boundaries (Shultz & Bale, 2006), morpho-phonology (Shultz, Berthiaume, & Dandurand, 2010), habituation of infant attention to auditory (Shultz & Bale, 2001, 2006) and visual (Shultz, 2011; Shultz & Cohen, 2004) information, false-belief (Berthiaume, Onishi, & Shultz, 2008; V.

C. Evans, Berthiaume, & Shultz, 2010), and concept acquisition (Baetu & Shultz, 2010; Shultz, Thivierge, & Laurin, 2008).

Our experience teaching university students about psychological development on the balance scale suggests that those few students who spontaneously use the torque rule to solve balance problems admit that they learned this method in science classes, either in secondary school or college. When the remaining students are informed that balance-scale problems can be solved by computing and comparing torques, they too begin to sometimes use this torque rule to produce more correct answers. Thus, it seems likely that most people learn a torque rule from explicit verbal instruction that includes relevant examples (Siegler, personal communication). In contrast, people are unlikely to learn a torque rule from examples alone because problems requiring the torque rule are so rare.

1.1.2 The new model

Here, we attempt to achieve a successful and psychologically more valid model of balance-scale development by capturing all four stages, including a genuine torque rule at stage 4. The new model combines and extends our initial balance-scale simulation (Shultz et al., 1994) and recent exploratory work with knowledge-based learning (Shultz et al., 2007).

We posit two different processing modules to solve the task, an intuitive module and a torque-rule module. The intuitive module is a connectionist network that implicitly learns environmental regularities about balance-scale problems. It predicts which side of a balance scale will tip down, without any need to invoke a rule. The learning process is essentially bottom-up and stimulus driven. In contrast, the torque-rule module simulates explicit learning via teaching a torque rule in secondary-school science classes. This rule has to be understood through language, and coded for in an appropriate manner in memory. In our model, we implement this as a neural module with torque-rule functionality. A meta-cognitive, selection module selects whether to use the prediction of the intuitive model or, if it determines that prediction is likely to be incorrect, invokes the torque module to receive a more definitive answer. All three modules are implemented as neural networks.

This sort of dual-processing approach (e.g., automatic vs. deliberate) is rooted in a long and currently active emphasis in cognitive psychology (J. B. T. Evans, 2010; Kahneman, 2011; Stanovich, 2012). Our results are discussed in that context.

2. Methods

2.1 Model architecture

Our model contains three key modules: intuitive, torque-rule, and selection. The intuitive and the torque-rule modules can compute separate predictions about the state of the scale balance after the beam is released from its supports. By learning to predict the correctness of the intuitive

module on various balance-scale problems, the selection module decides whether to use the prediction of the intuitive module or invoke the torque module instead.

All three modules are implemented using cascade correlation variants. More specifically, the intuitive and selection modules uses sibling-descendant cascade correlation (SDCC) (Baluja & Fahlman, 1994), whereas the torque module employs knowledge-based cascade correlation (KBCC) (Shultz & Rivest, 2001). We first describe these variants, and then their use.

2.2 Cascade correlation

Cascade correlation (CC) is a neural network algorithm characterized by network expansion as needed to solve some problem. This automated growth approach solves the ill-defined problem of selecting an appropriate topology of hidden units (number of units and their arrangement in layers) as used in feed-forward neural networks.

CC learns by alternating between two phases: input phase and output phase (Fahlman & Lebiere, 1990). CC always begins in output phase with the simplest possible network, that is, one without any hidden units. In output phase, CC learns by adjusting connection weights entering output units using a standard gradient descent on output error. If the current topology does not allow a sufficient error reduction, CC shifts to an input phase at the end of which a new hidden unit is recruited from a pool of candidate units. In input phases, connection weights between inputs and these candidate hidden units are trained so as to maximize the covariance between unit activation and the residual network error. At the end of an input phase, the candidate unit with the highest absolute covariance is selected and installed into the network with random input connection weights of the same sign as just learned, the other candidates are discarded, and there is a shift back to output phase. The algorithm shifts from one phase to the other when the current phase fails to improve the solution of the problem on which the network is being trained, by not reducing error or failing to improve covariances, for output- or input-phase, respectively. Alternation between phases typically continues until network error is sufficiently low, or a maximal network size is reached, the size of the network increasing by one hidden unit at the end of each input phase.

Two important parameters characterize variants of this algorithm: (1) where to install newly recruited hidden units in the existing network, and (2) what kind of computation these hidden units perform. Regarding installation location for hidden units, solutions vary from flat topologies in which all units are installed on a single hidden layer (Sjogaard, 1992) to deep topologies in which all recruited units are cascaded on new, progressively deeper layers (Fahlman & Lebiere, 1990). The following two sub-sections detail how algorithm variants deal with these two parameters.

2.2.1 Sibling-descendant cascade-correlation

In our present model, we use a compromise approach called the sibling-descendant cascade-correlation (SDCC) (Baluja & Fahlman, 1994) which flexibly installs a new recruit on the

current deepest layer, or on a new, deeper hidden layer. Apart from network depth and number of connection weights (Shultz, 2006), the choice of installation strategy has little impact on functionality. A systematic study of the effect of cascading weights and network depth showed that flat and deep variants of generalize equally well (Dandurand, Berthiaume, & Shultz, 2007). Mathematical details about CC and SDCC are available elsewhere (Shultz & Fahlman, 2010).

2.2.2 Knowledge-based cascade correlation

Regarding the kind of computations that hidden units perform, ordinary hidden units have sigmoidal activation functions (Fahlman & Lebiere, 1990) to compute output values given the weighted sum of input values to the unit. Knowledge-based cascade correlation (KBCC) generalized the concept of recruits to any differentiable function, including previously-learned CC or SDCC networks (Shultz & Rivest, 2001) or human-designed networks or units that have some symbolic-like functionality (Shultz, Rivest, Egri, Thivierge, & Dandurand, 2007). The computational device that gets recruited is the one whose output covaries best with residual network error. A simple example of a KBCC network is shown in Figure 1, illustrating that a recruited source network or function can have multiple inputs and outputs, thus requiring connection-weight matrices rather than vectors. Mathematical details about KBCC are available elsewhere (Shultz & Rivest, 2001; Shultz, et al., 2007).

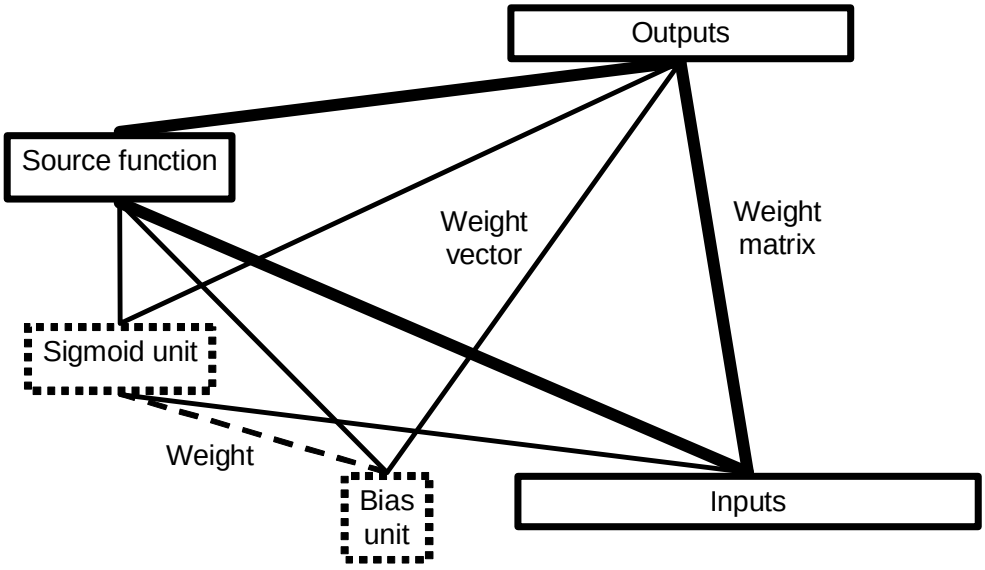

Figure 1. Drawing of a sample KBCC network that has recruited a single sigmoid hidden unit followed by a source function. Thick solid lines represent connection-weight matrices, thin solid lines represent connection-weight vectors, and the dashed line represents a single connection weight.

2.3 Intuitive module

As in our initial CC simulation of the balance scale task (Shultz et al., 1994), the intuitive module learns to predict balance-scale results from learning with examples only, and is implemented here as an SDCC network. The recruitment pool contains eight sibling and eight descendant units that all have sigmoidal activation functions whose outputs range from -0.5 to 0.5. The intuitive module receives four inputs representing distance and weight on the right and distance and weight on the left. There are two outputs, whose target patterns are coded as follows: +0.5 +0.5 for balance, +0.5 -0.5 for left heavier, and -0.5 +0.5 for right heavier. This improved choice of coding values for the balance scale uses target values in the stable, saturated regions of the activation function, yielding somewhat faster learning. In the original CC simulation (Shultz, Mareschal, et al., 1994), the target values for a balanced outcome were 0 0, which reside in the steep, transitional range of the activation function.

Training begins with 100 initial patterns, randomly selected from the 625 possible balance-scale problems allowed by five weights and five distances from the fulcrum. In the selection process, there is a .9 bias toward equal-distance problems (in which the weights are placed equally distant from the fulcrum). This is to encourage early use of the weight rule (the side with more weights should descend) under the assumption that children have rather few experiences with physical devices that systematically vary distance from a fulcrum (McClelland, 1989). One new pattern is added in each output epoch, under this same .9 bias. In an epoch, each of the training patterns is encountered once. Because items are selected with replacement, random selection of duplicate patterns is permitted.

Exploratory simulations indicate that these networks are well into stage 3 by about 350 epochs (see confirming evidence in Results section). Thus, when a network reaches 350 epochs, we allow it to complete the current output phase, and then stop training. Thus, training stops after a fixed period regardless of the level of error that was reached. All other parameters are the default values.

2.4 Torque module

After the intuitive module is trained, our simulation proceeds to training the torque-rule module. This involves a fresh KBCC network and training data that includes the intuitive training set plus perhaps an infusion of torque problems (see *practice* factor below). We manipulate two variables, in a two-way independent-factors design. First, a *teaching* factor represents whether or not the torque rule is injected into the KBCC candidate pool (see details below). Second, a *practice* factor represents whether or not the intuitive training set is expanded with a randomly selected 26 of the 52 possible torque problems. The other 26 torque problems are reserved for testing generalization. This design allow us to systematically study the effect of teaching the torque rule and giving additional practice with torque problems, mimicking what transpires when torque is covered in secondary-school science classes.

The torque module uses KBCC. In the recruitment pool, eight sibling sigmoid and eight descendant sigmoid units are provided, and in addition eight sibling torque rules and eight descendant torque rules for the conditions in which teaching is occurring. The target torque-rule network has the same inputs and outputs as the intuitive network.

To allow networks sufficient opportunity to fully exploit the error reduction possible with a complex unit like the torque rule, we allowed input and output phases to be longer by setting the patience parameter to 100 epochs, and the change threshold to 0.001. Training begins in input phase rather than the usual output phase, and ends when a standard score threshold criterion of .4 on network error is met. Other simulation parameter settings are the same as for the intuitive network.

2.4.1 Torque-rule injection

To simulate the teaching of a torque rule, we introduce into the recruitment pool a unit (hereafter referred to as the torque rule) which executes the following function on its four inputs:

$$TR = \frac{1}{1+e^{-4TD}} - 0.5 \qquad \text{Equation 1}$$

where $TD = (w_r d_r) - (w_l d_l)$ Equation 2

Here, *TR* is the torque rule, and *TD* is torque-difference, computed as the difference between the torque on the right side of the fulcrum and the torque on the left side of the fulcrum. On each side of the fulcrum, torque is computed as the product of weight (*w*) and distance (*d*). *TD* is then passed through a sigmoid squashing function to obtain *TR*, as shown in Figure 2 for points corresponding to discrete values of weight and distance. *TR* is also a differentiable function, which KBCC requires of potential recruits. The exponent of 4 increases the steepness of TR, emphasizing the binary judgments that humans are asked to make on this task, but the reported results were also produced with the default exponent value of 1.

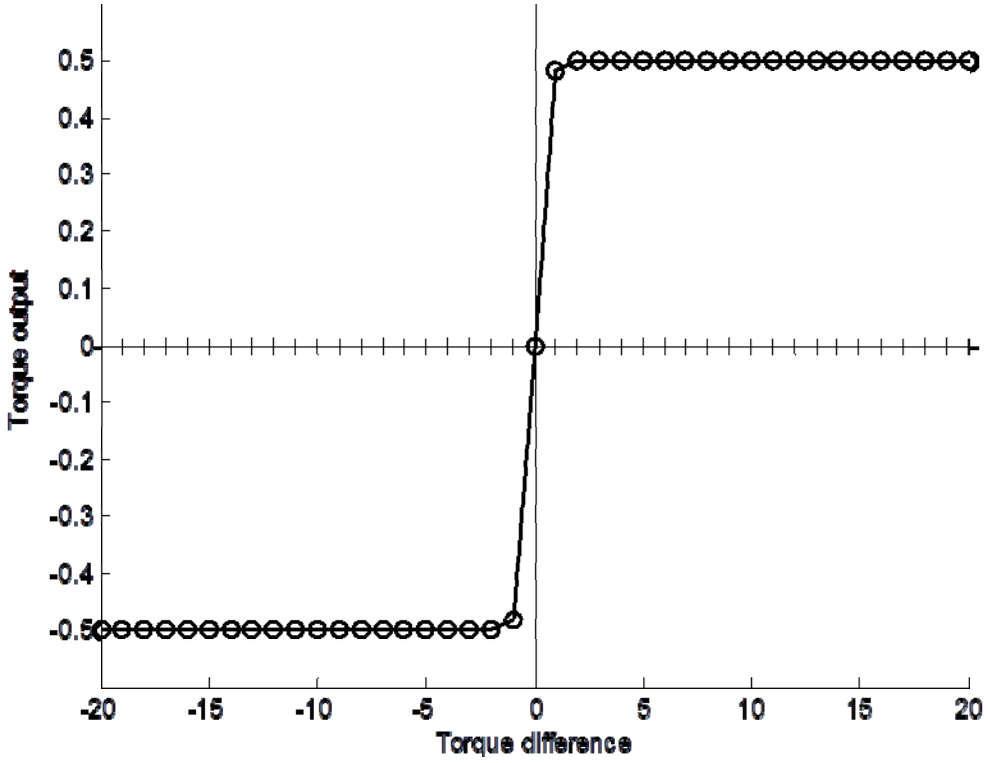

Figure 2. Torque-rule output as a function of torque difference.

2.5 Selection module

The selection module learns to predict the accuracy of the intuitive network and then decides, for each balance-scale problem, whether to use the intuitive response or invoke the torque-rule module. The selection module can thus be interpreted as a meta-cognitive system that monitors correctness of the intuitive module and decides whether to activate the torque-rule.

Implemented as an SDCC network, the selection module receives seven inputs. In addition to the usual four inputs describing a balance-scale problem, these include a torque-difference measure (the absolute value of Equation 2) and two binary inputs indicating symmetry of weights and symmetry of distances (1 if symmetrical, 0 otherwise). These additional inputs, which presumably can be easily extracted perceptually in humans, provide useful information in this rapid, heuristic estimation task. The training set consists of the balance problems that were presented to the intuitive network. The output target is the correctness value of the intuitive network on these problems (correct = 1, error = 0). The output range of sigmoids is selected to be between 0 and 1. Parameter settings are all default values.

After the selection module is trained, the system's prediction is generated according to a simple rule: if the intuitive module is expected to give a correct answer, then output the value it predicts, otherwise output the prediction of the torque module.

2.6 Test Sets

The system is tested with three different sets of problems, labelled Siegler-TD, Addition, and Torque. We decided not to additionally test performance with Latent Class Analysis (LCA) because of its demonstrated tendency to create small unreliable rule classes and inability to decide on the right number of classes, and because results were quite clear with the tests we did perform. Appendix C demonstrates these first two LCA limitations.

2.6.1 Siegler-TD

The so-called Siegler-TD test set contains 24 balance-scale patterns selected as in our original simulation (Shultz, Mareschal, et al., 1994), inspired by Siegler's (1976) test set but additionally balanced for torque-difference effects. It contains four randomly-selected problems of each of Siegler's six types: balance, weight, distance, conflict-balance, conflict-weight, and conflict-distance problems. Except for balance and conflict-balance problem types that always have a torque difference of 0, other types of problems are represented at four different levels of torque difference: 1, 3-5, 6-9, or 10-19. This is an improvement over studies that ignore torque differences and thus risk confounding problem type with torque difference and studies that use only small torque differences and thus risk underestimating torque-difference effects.

This test set is used to diagnose stages 0-4 according to Siegler's (1976) criteria, with the proviso that Stage 2 is given diagnostic priority over Stage 3 (Shultz, Mareschal, et al., 1994). Rule diagnosis is conducted by software: diagnosis of Stage 4 requires 20 of 24 problems correct; diagnosis of stage 2 requires at least 13 correct on the 16 balance, weight, distance, and conflict-weight problems and less than 3 correct on the 8 conflict-distance and conflict-balance problems; stage 3 requires at least 10 correct on the 12 balance, weight, and distance problems and fewer than 10 correct on the 12 conflict problems; stage 1 requires at least 10 correct on the 12 balance, weight, and conflict-weight problems and fewer than 3 correct on the 12 distance, conflict-distance, and conflict-balance problems. Stage 2 is given scoring priority over Stage 3 because the criteria for Stage 2 are more specific, particularly on how to score conflict-weight problems.

2.6.2 Addition

The addition test set helps to distinguish a genuine torque rule from a mere addition rule. It contains all 148 addition problems (among all conflict problems), a few of which may be included in the training set when expanding by one pattern per epoch. Typically, no more than one or two such patterns get included in the train set.

2.6.3 Torque

The torque test set contains the 26 torque problems not randomly selected to expand the training set in torque-rule training. Recall that among all conflict problems, there are 52 torque problems,

half of which are used in training. There is a small probability that these problems are selected when expanding the train set by one pattern per epoch, but in practice no more than a single pattern is included in this way. If a network performs well on torque problems, then it is diagnosed as following a genuine torque rule as opposed to solving balance-scale problems with the often successful addition rule.

2.7 Test for growth spurts

We tested for growth spurts in the acquisition curves with Automatic Maxima Detection (AMD). Such spurts often signal the transition between successive stages, which can be indicated by plateaus. AMD uses functional data analysis to distinguish statistically significant spurts from continuous development and mere noise (Dandurand & Shultz, 2010).

3. Results

We ran 20 models, with each model containing randomly-selected, distinct training sets and instances of the three modules just described.

3.1 Intuitive module

Figure 3 presents mean stage classification on the Siegler-TD test set for 20 intuitive networks over epochs. Performance at stage 1 is evident around epoch 30-40, stage 2 at epochs 100-150, and stage 3 at epochs 200-350. Epoch 50 marks the transition between stages 1 and 2, and epoch 150 marks the transition from stage 2 to 3. Thus, these networks capture the first three balance-scale stages seen in children from about five years of age up through early adolescence.

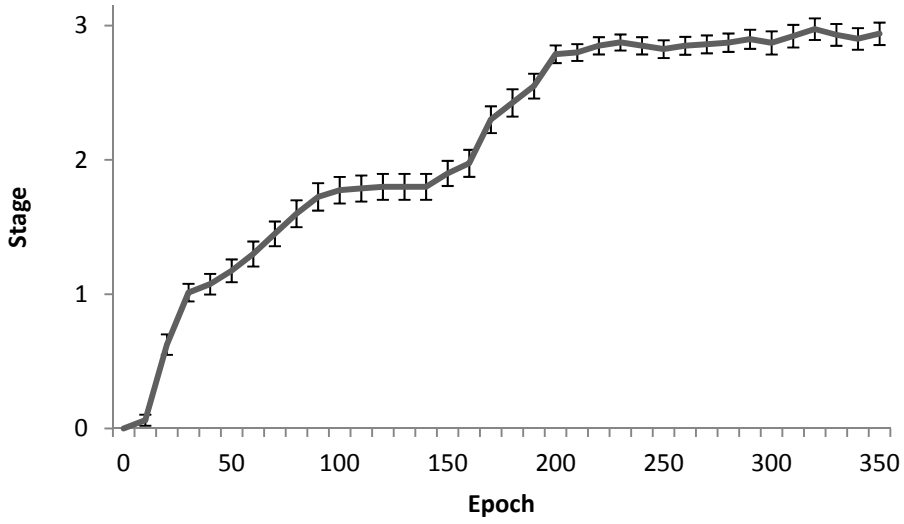

Figure 3. Stage progression of models on the Siegler scale as a function of training, with standard errors.

We then used AMD to determine if stage transitions were characterized by significant spurts in stage progression. Results, shown in Figure 4, show that the transition between stages 2 and 3 is characterized by a significant, $p < .05$, performance spurt with maximal velocity at epoch 177. AMD identifies spurt locations by local maximal velocity (first derivative), decreasing acceleration (second derivative), and negative jerk (third derivative) (Dandurand & Shultz, 2010). The lambda parameter in AMD controls the amount of smoothing that is applied. We used a lambda of $1x10^8$.

The present result is consistent with visual inspection of Figure 3 where the clearest plateaus occur at stages 2 and 3, with a marked increase in performance between these plateaus. In contrast, progression from stages 0 to 2, although bumpy, proceeds in a more gradual and steady fashion. Using different methodology, a back-propagation neural network model was also shown to progress from stage 1 to 2 of the balance scale in a continuous fashion (Schapiro & McClelland, 2009). That model has not been shown to reach stage 4 by any measure.

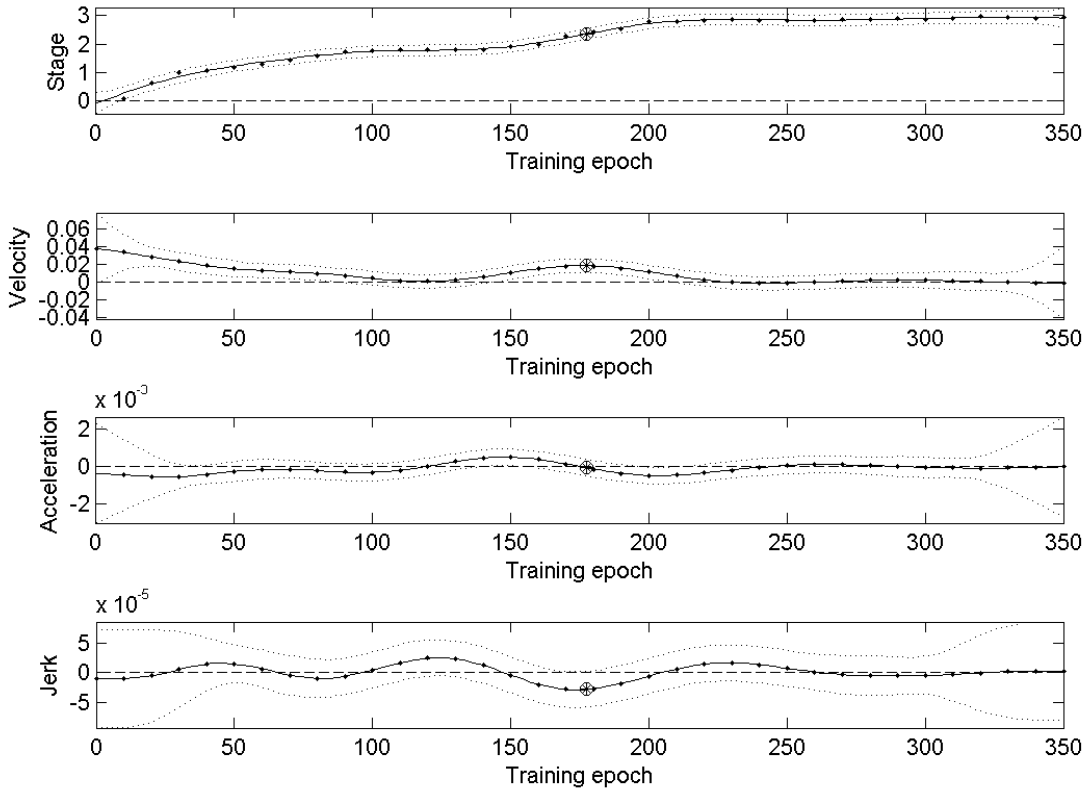

Figure 4. AMD analysis of the stage progression with lambda = $1x10^8$ and p = .05

Figure 5 shows accuracy, in terms of mean proportion correct, in 20 intuitive networks on each of the three test sets over epochs. This confirms that intuitive networks learned to perform

well on the Siegler-TD and addition test sets, but not on the torque test set. Model accuracy is consistently higher on the addition test set than on the torque set. In addition, around epochs 150-200, we observe a rapid increase in accuracy on addition problems combined with a decrease of accuracy on torque problems. This strongly supports the hypothesis that networks develop an addition strategy, and that addition characterizes stage 3 performance as argued by a number of researchers in developmental psychology (Boom, et al., 2001; Ferretti, Butterfield, Cahn, & Kerkman, 1985; Jansen & van der Maas, 1997, 2002; Normandeau, Larivee, Roulin, & Longeot, 1989; Quinlan, et al., 2007).

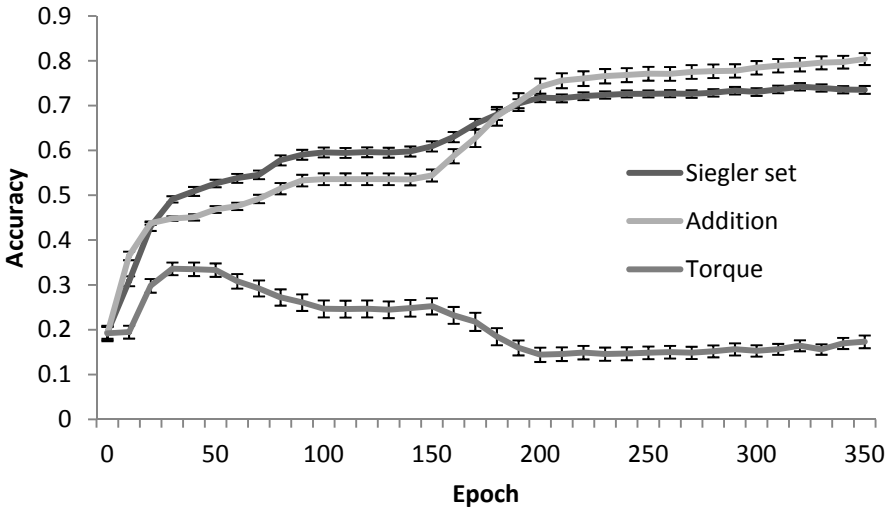

Figure 5. Average accuracy on Siegler, addition and torque problems as a function of model training, with standard errors.

We next investigated accuracy as a function of torque difference. Figure 6 shows that the current model successfully replicated the torque-difference effect previously found (Shultz, Mareschal, et al., 1994) over the 4 torque levels (level 4=high, 1=low). This effect, also observed in children (Ferretti & Butterfield, 1986; Ferretti, et al., 1985), describes the fact that accuracy tends to be better for problems when the torque difference between the two sides of the scale is large.

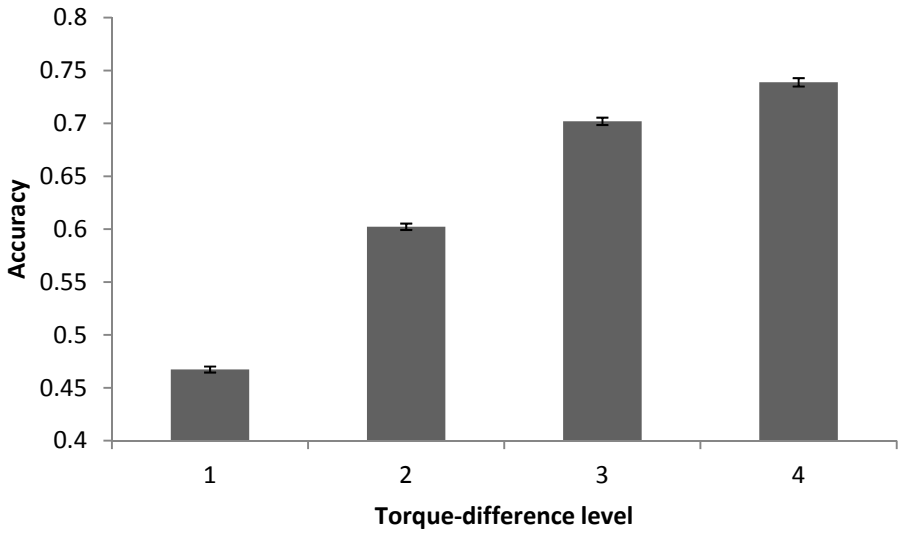

Figure 6. The torque-difference effect in the intuitive network of the balance scale task, with standard errors**.**

3.2 Torque module

Figure 7 shows accuracy of the torque module on balance-scale problems as a function of training. The four conditions correspond to the 2x2 independent-factors design: (1) *teaching* indicates whether the torque rule is available as prior knowledge for recruitment (2 levels = present or absent) and (2) *practice* indicates whether the training set was expanded with half of torque problems to give additional, torque-problem-specific practice (2 levels: expanded or not). Both teaching and practice are necessary for achieving high performance (about 0.9 accuracy) on torque problems. We also see that the combination of teaching and practice results in a higher accuracy on all three data sets.

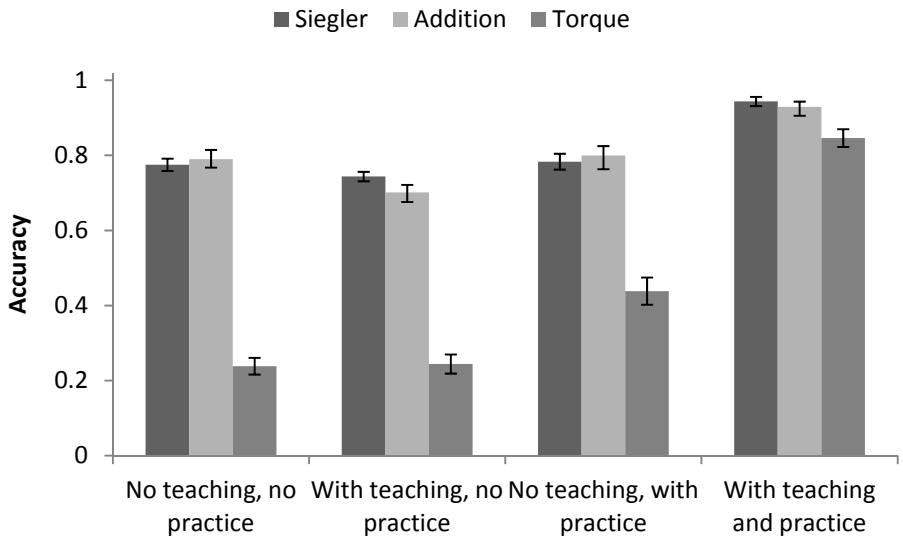

Figure 7. Accuracy of the torque module on three types of test problems, as a function of teaching the torque rule and of additional torque practice problems.

In conditions that included injection of the torque rule, networks recruited between 1.9 and 2.1 torque rules, but no sigmoid unit. In contrast, for conditions in which the torque rule was not available, between 3 and 7 sigmoid units were recruited.

3.3 Selection module and the complete model

The selection networks trained for a mean of 443 epochs (SE = 33) and recruited 1.3 (SE = 0.1) hidden units on average.

Figure 8 presents mean global accuracies of the model on the three test sets, comparing the intuitive module alone with the combination of intuitive and torque answers as selected by the selection module and with an idealized symbolic torque rule. Although intuitive networks perform well on the Siegler-TD and addition test sets, they do badly on torque problems. In contrast, the combination strategy yields good performance on all three test sets, qualifying as stage 4 performance as described by Siegler (1976), requiring success rate of at least .8. This result is robust whether we consider the connectionist or the symbolic torque rule module implementation.

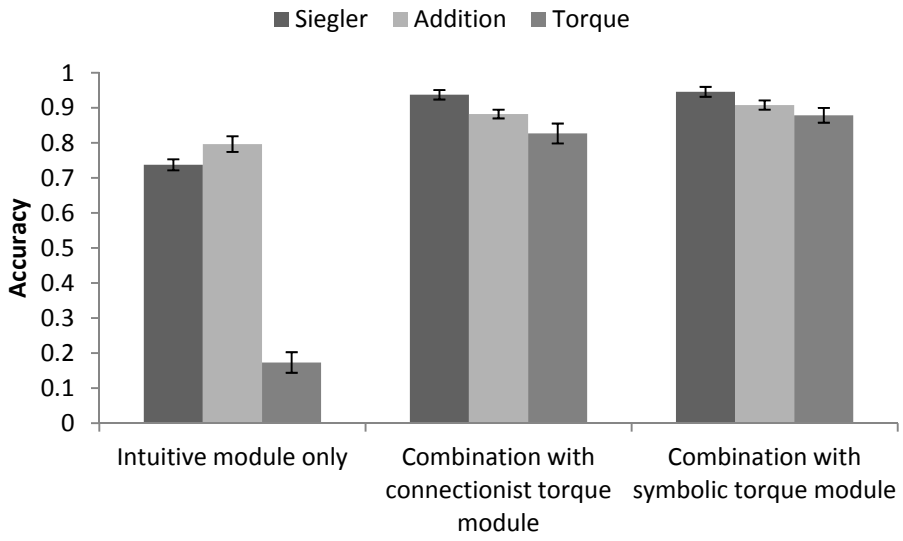

Figure 8. Comparison of accuracies of the intuitive module only and the combination model (either with a connectionist or a symbolic implementation of the torque module) on Siegler, addition and torque problems, with standard errors.

The mean proportions of problems solved by the intuitive network under combination conditions were .75 (*SE* = 0.01), .64 (*SE* = 0.01), and .25 (*SE* = 0.03) on the Siegler-TD, addition, and torque test sets, respectively. That is, torque problems are less likely than other problems to be solved intuitively. We then further investigated which problems get solved by the intuitive network and by the torque network. As we can see in Figure 9, simple problems (balance, distance and weight) tend to be solved more often using the intuitive network module, while conflict problems tend to be solved using the torque network module.

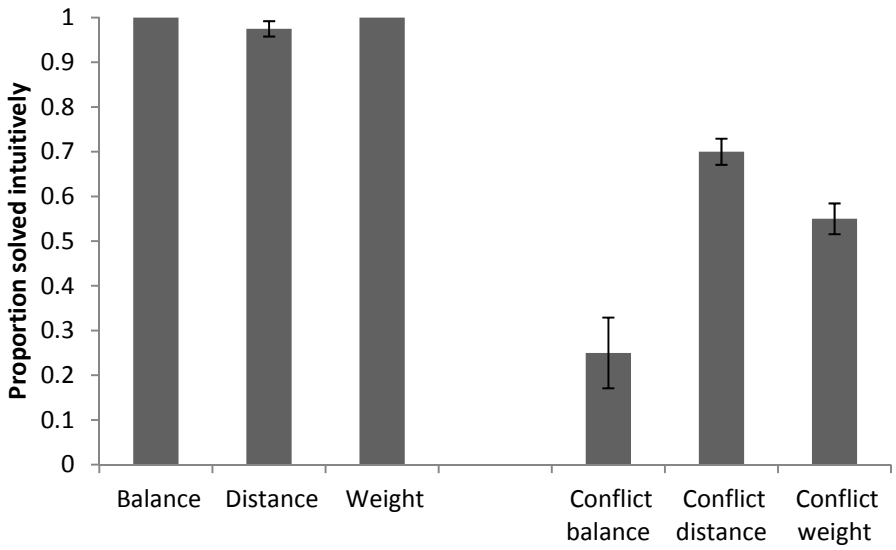

Figure 9. Proportion of problems solved using the intuitive network in the combination model. Results are presented by problem type for the Siegler test set.

We can next make some predictions on response times of children as they solve the six kinds of problems in the Siegler test set. Figure 10 presents possible timings of execution of the three modules (selection, intuitive and torque rule) for different combinations of parallel and serial processing, under three related assumptions. First, the selection module always executes first to decide which answer (intuitive or torque rule) to use. Second, the default or preferred mode of processing of the system is intuitive. Third, processing with the torque module will only be executed when the selection module predicts that the answer of the intuitive system is likely to be incorrect.

As we can see in Figure 10, the model predicts slower response times on problems solved with the torque rule module than problems solved with the intuitive module, as long as the time needed to generate an answer using the torque rule module (i.e., t_1) is longer than the time needed to generate an answer with the intuitive module (i.e., t_2). If $t_{tr} > t_i$, this constraint is satisfied.

For the Siegler test set, the model predicts slower response times for conflict problems than for simple problems because answers to conflict problems rely much more on the torque module than answers to simple (non-conflict) problems. Quantitative details of such predictions depend on the relative ratio of time spent in the two modules, on the relative speed of processing of the modules, and on the characteristics of the processing (parallel vs. serial). However, as long as the torque rule module (t_{tr}) is slower than the intuitive module (t_i), the model robustly predicts slower responses on problems solved with the torque rule. In the example shown in Figure 11, based on the proportions of problems solved intuitively presented in Figure 9, processing times of the torque module is twice as long as that of the intuitive module ($t_{tr} = 2t_i$). This predicted

pattern of response times is compatible with the response times observed in young adults, plotted in Figure 12 (van der Maas & Jansen, 2003).

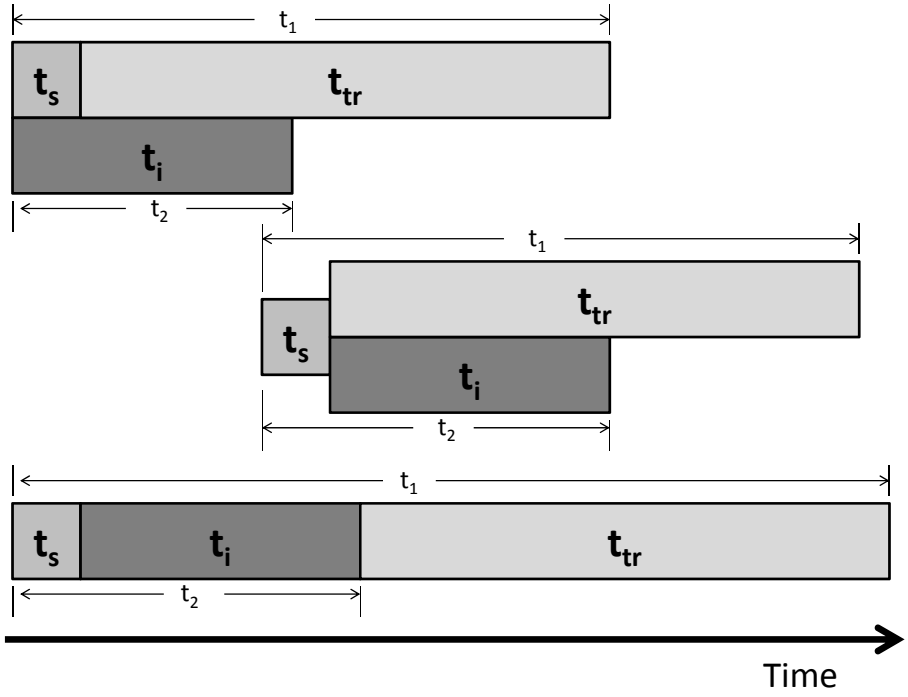

Figure 10. Three possible modes of execution of the processing modules, with t_s, t_i and t_{tr} corresponding to the processing time of the selection, the intuitive and the torque-rule modules, respectively. Note that the model presented on the middle row covers two cases of execution of the intuitive and the torque rule modules: (1) they are executed in parallel, and thus both answers are generated though only one is used, and (2) only the module corresponding to the answer to be used is executed. For the model on the lowest row, the intuitive module always executes, even when its answer is not used. In all three cases, t_1 is the response time for items solved using the torque rule module, and t_2 the response time for items solved using the intuitive module.

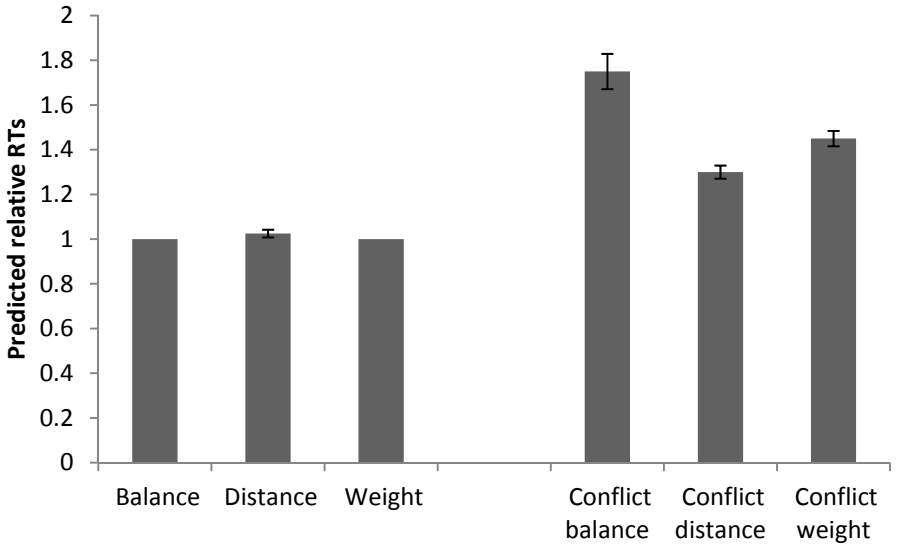

Figure 11. Illustration of predicted response times as a function of balance-scale problem type, with standard errors.

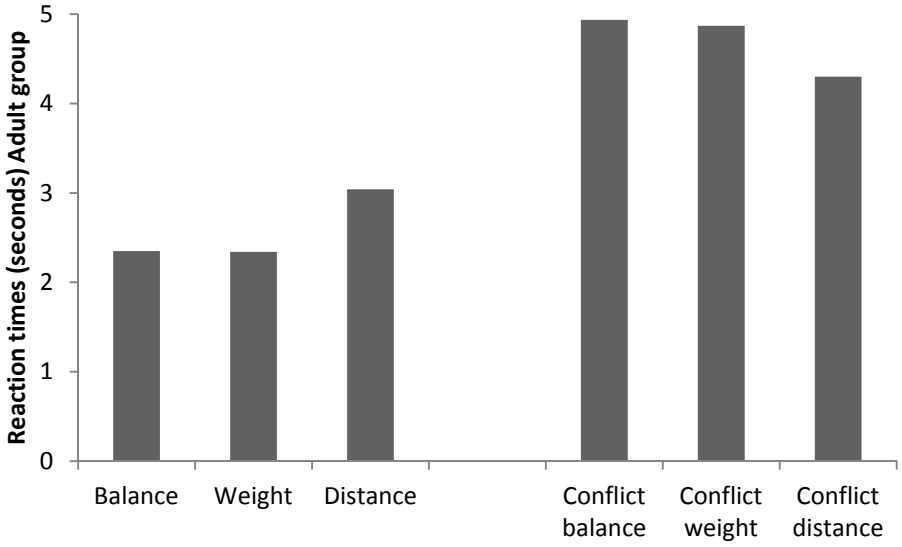

Figure 12. Human response times on balance scale problems classified according to Siegler's scheme. Data for this graph are from Table 3 in van der Maas and Jansen (2003). The value for the conflict balance was computed as the average of times in conflict balance of types A and B.

4. Discussion

4.1 Summary and interpretation of findings

Our new model of development of balance-scale knowledge progresses through all four balance-scale stages, including a third stage in which weight and distance information are added and a

fourth stage in which these two sources of information are multiplied as in a genuine torque rule. Coverage of a genuine torque rule contradicts previous doubts about the inability of constructive neural networks to achieve that level of performance (Quinlan, et al., 2007). A torque rule can be considered genuine if it generalizes well to problems that cannot be solved by a simpler rule that merely adds, rather than multiplies, weight and distance information. Our networks generalize correctly to such untrained torque problems with about 90% success. This is not perfectly correct performance, but then neither is that of human participants on these tasks.

This model incorporates three constructive neural networks. It captures the first three balance-scale stages with an intuitive network that learns only from examples. Then a knowledge-based network with an injected torque rule in the source-knowledge pool and additional torque training examples builds on this early intuitive training by recruiting this taught torque rule and learning how to use it. This model is the only neural-network system to so far demonstrate progression through all four balance-scale stages finishing with a genuine torque rule. Using automatic maxima detection (AMD), we found that the transition from stage 2 to 3 in the intuitive networks was the only significant spurt; earlier transitions were continuous. A symbolic rule-based model also ends with a genuine torque rule (van Rijn, van Someren, & van der Maas, 2003), but the ordering of stages 1 and 2 and the late appearance of addition and torque rules in that model were engineered by parameter settings; they were not an emergent result of learning or development.

Operation of our intuitive network covers the torque-difference effect by showing better accuracy on problems with high absolute torque difference from one side of the scale to the other. This simulates psychological evidence (Ferretti & Butterfield, 1986; Ferretti, et al., 1985) and replicates our initial simulation (Shultz, Mareschal, et al., 1994). The symbolic rule-based model showed a torque-difference effect only with respect to differences in distance but not weight and only in the vicinity of stage transitions (van Rijn, et al., 2003), not throughout development as children apparently do.

Just as with secondary-school science students, a lesson on torque does not guarantee a torque solution. The taught torque rule must be stored, recruited, and practiced; and even then it is not required to solve simple balance-scale problems that can be solved intuitively. In our model, a selection network learns to predict whether the intuitive network is likely to be correct on any given balance-scale problem. If the prediction is negative, the torque-rule network is invoked for a more accurate answer.

Results presented in Appendix A confirm our earlier conjecture (Shultz & Takane, 2007) that even SDCC networks learning solely from examples can acquire a genuine torque rule, if enough of those examples can only be solved by comparing torques. The training patterns there contained equal numbers of addition and torque problems. ANOVA results show that these networks learn an addition rule early and a torque rule later on, a natural progression for

networks that sum their inputs and recruit nonlinear hidden units as required. LCA results in Appendix B confirm this rule progression.

This again contradicts claims that constructive neural networks cannot cover a genuine torque rule (Quinlan, et al., 2007). However, because those SDCC networks do not progress through the first two stages of balance-scale performance seen in children (using weight and weight plus distance), they are not our favored comprehensive balance-scale model. Progression through these first two stages requires a training set with a strong bias in favor of equal-distance problems in which there are the same numbers of weights placed equally distant from the fulcrum. This bias, coupled with the inherent rarity of torque problems fails to provide sufficient experience to build a torque rule.

This explains why ordinary folks, unlike SDCC networks or Archimedes, do not discover a torque rule on their own. See Shepard (2008) for a clever thought experiment on how Archimedes might have discovered the nature of torque from reasoning alone (c. 280 BC). Most of us would need to be taught how torque works and then get some practice applying it (Siegler, personal communication). Our favored model of the balance scale works in that same fashion. Our experiments confirm that a combination of teaching (implemented by an injected torque rule) and practice is required for torque-like performance, after progressing through earlier stages of weight, weight plus distance, and addition.

Our model correctly predicts that the torque-rule module is more likely to be invoked on the relatively difficult conflict problems than on simple problems that present no conflict between weight and distance information. On the simple assumption that an intuitive solution is generated faster than a torque-rule solution, the model also covers psychological evidence that response time is slower on conflict than simple problems (van der Maas & Jansen, 2003). The more effortful, deliberative process is not invoked until it is needed, i.e., when the intuitive solution is expected to be incorrect.

On this same basis, our model also predicts, so far uniquely, that response times would increase on problems with small absolute torque differences between the sides of the scale. This again is because the torque-rule module tends to be used more often when torque differences are small. In fact, for all the non-balanced problems in the Siegler test set (weight, distance, conflict-weight, and conflict-distance problems), we find a point-biserial correlation between absolute torque difference and reliance on the intuitive network, $r(318) = .47$, $p < .001$. This correlation is even larger when only non-balanced conflict problems (conflict-weight and conflict-distance problems) are included, $r(158) = .73$, $p < .001$. On the well-supported assumption that intuitive solutions are faster than deliberative solutions, these correlations represent a relation between reaction time and torque difference – faster response to problems with a larger absolute torque difference. In humans, this prediction could be tested by computing the same correlations. If the prediction is confirmed in humans, it could be interpreted as relatively quick solutions to problems with small torque differences, on which it is easy to gain an intuitive impression of

what will happen when the beam's supports are removed, and longer response times to problems with such small torque differences that the torque rule must be invoked.

The inherent ability of KBCC to incorporate differentiable functions into its source knowledge pool is a novel and promising way to integrate neural-network and symbolic approaches to cognitive modeling. Our use of a confidence network indicates that neural networks also may be able to simulate aspects of meta-cognition or reflection. Although we implemented a torque rule in thoroughly neural fashion, there is no current evidence for how torque processing is done in people. We showed that our overall model can accommodate other implementations of a torque module that provide good performance on torque problems, as evidenced by the robustness of the pattern of results obtained using an optimal symbolic rule-processing implementation. It may be overkill to implement a whole rule-based system for a single rule. Also, it seems reasonable to suppose that brains must ultimately perform computations by learning how to pass activation signals among neurons.

Nonetheless we remain open to other possible implementations of rule-like processing. For instance, efficient computation of products can be accomplished with so-called sigma-pi units that multiply, rather than add, their inputs (Durbin & Rumelhart, 1989). Do this twice and compare the size of the products, and you have a torque rule. Also, a model called long short-term memory (LSTM) provides neurally-plausible modules of memory buffers and gates to retain representations over time (Hochreiter & Schmidhuber, 1997). Such features appear useful for a neural implementation of symbolic processing. For now, it is worth noting that our overall results and conclusions would remain consistent as long as implementation of the torque rule performs correctly on torque problems.

Our present model builds on, improves, and extends work presented earlier (Dandurand & Shultz, 2009). From that previous model, here we improve the coding of balance outcomes by not placing target outputs at the inflection points of sigmoid activation functions; use a selection network rather than a less-efficient confidence network to guide processing; extend the model to cover the torque-difference effect and reaction times; apply automatic maxima detection to detect growth spurts; isolate the effects of learning and practicing the torque rule; add appendices to report on learning from examples alone (A), diagnosing stages with LCA and RAM techniques (B), and demonstrating reliability and decidability problems with LCA (C); and, in the next section, relate our model to previous work on dual and tripartite processing.

4.1.1 Relation to previous work

As noted, our modular approach to balance-scale phenomena is consistent with a long and still active emphasis in psychology. This work has been variously described in many different contrasts, such as heuristic vs. analytic (J. B. T. Evans, 2003, 2006, 2010), implicit vs. explicit (Reber, 1989; Reber & Lewis, 1977), automatic vs. controlled (Shiffrin & Schneider, 1977) or conscious (Posner & Snyder, 1975), adaptive vs. conscious (Wilson, 2002), intuitive vs. reasoned (Kahneman & Frederick, 2005), associative vs. rule-based (Sloman, 1996), and fast vs. slow

(Kahneman, 2011). Perhaps because of this proliferation of semantically rich terminology, some have opted for more neutral, numerical terms such as types 1 and 2 (J. B. T. Evans & Wason, 1976; Stanovich, 2012; Thompson, Prowse Turner, & Pennycook, 2011; Wason & Evans, 1975) or systems 1 and 2 (Kahneman, 2011).

A few researchers are proposing addition of a third, metacognitive module that decides whether to accept the response submitted by the faster, intuitive module or to inhibit that response and activate the more deliberate reasoning module for a more deeply considered response (Stanovich, 2012; Thompson, et al., 2011). Some of this work on dual (Sun, Slusarz, & Terry, 2005) or tripartite (Hahn & Nakisa, 2000) systems has involved operative computational models, while other researchers have begun to explore, in some cases with the aid of computational models, the brain characteristics that might have led to these differences in processing (Frank, Cohen, & Sanfey, 2009).

Our current model seems closest to that of Hahn and Nakisa (2000) on plural inflection of German nouns. Their model coupled a neural network with a single default rule to construct 15 different ways to form the plural form of 4000 German nouns. If the strength of the memory for an exception noun rose above a certainty threshold, default rule application was blocked. But if memory strength remained below this threshold, the default rule of adding an *–s* to the singular form was applied. This is similar to our model in having three modules implementing a rule, a network, and a selector, but the results were quite different. In their system, performance was always more correct when the default rule was not applied, regardless of threshold level, thus contradicting a dual-process hypothesis. In contrast, our model used the torque rule to improve performance on the more difficult problems where intuitive decisions were not so obvious. Such variations in results may naturally correspond to task differences. For the German plural, the rule turns out to be superfluous because the network can handle this default case as well as the 14 exceptional forms. Whereas, for the balance scale, some problems have such similar torques that a superficial, intuitive solution is too error-prone to be trusted.

Another difference is that deciding whether to use the default plural rule was done exclusively in symbolic software, whereas our selection module contained a neural network to learn to monitor performance of the intuitive module. It is not widely appreciated that neural networks can serve a meta-cognitive role by predicting the performance of another neural network rather than events in the environment.

5. Conclusion

In conclusion, we present a comprehensive model of the balance-scale task that successfully simulates the stage progressions, including the recently disputed torque-rule performance in the terminal stage, as well as the torque-difference effect and the pattern of response times observed in children. The model posits two distinct processing modules, an intuitive module and a torque-rule module, consistent with the well-supported dual-processing approach in cognitive

psychology. A third, meta-cognitive module determines which of these two modules will produce the answer to any particular balance-scale problem. In the model, stage progression occurs as a natural consequence of learning (due to connection-weight adjustment) and development (due to recruitment of hidden units) in a fully connectionist fashion. Although the model uses a connectionist implementation of an explicitly-taught torque rule, it can accommodate any implementation that provides equivalent torque-comparison functionality, including symbolic versions. As with meta-cognition, it is not widely appreciated that neural networks can implement symbolic functions.

Acknowledgements

This work is supported by a post-doctoral fellowship to FD and an operating grant to TRS, both from the Natural Sciences and Engineering Research Council of Canada. We are grateful to Yoshio Takane for discussions of LCA, some data analyses, and comments on earlier drafts.

Appendix A: Simulating Rule 4 in Balance-Scale Development

A.1 Introduction

A recent critique argued that connectionist models of balance-scale development do not capture stage-4 performance (Quinlan, van der Maas, Jansen, Booij, & Rendell, 2007), which on some theoretical accounts (Siegler, 1976) involves computation and comparison of torques. Torque is the rotational force applied to a lever, multiplied by its distance from the lever's fulcrum. Because many conflict problems can be solved by adding (rather than multiplying) weight and distance, documentation of a torque rule needs to be supported by success on problems that cannot also be solved by addition.

With five pegs and five weights, the problem size used in our original cascade-correlation (CC) simulations of the balance scale (Shultz, Mareschal, & Schmidt, 1994) and in many balance-scale psychology experiments, there are 625 total problems, of which just 200 are conflict problems. Only 52 of these conflict problems (dubbed *torque* problems) actually require a torque rule for correct solution because the other 148 conflict problems (dubbed *addition* problems) can be solved correctly by adding distance and weight on each side and comparing these sums.

Until recently, there was not much interest in using an addition rule to gage balance-scale performance and addition did not figure importantly in descriptions of balance-scale stages. Thus, addition was routinely ignored in both symbolic (Langley, 1987; Newell, 1990; Schmidt & Ling, 1996) and connectionist (McClelland, 1989; Schapiro & McClelland, 2009; Shultz, et al., 1994) simulations of balance-scale performance, and there was no attempt to distinguish addition from a genuine torque rule in these simulations.

Thus it is not surprising that researchers using sophisticated methods such as Latent Class Analysis (LCA) failed to find evidence of a torque rule, distinct from an addition rule, in replications of connectionist models (Jansen & van der Maas, 1997; Quinlan, et al., 2007). However, the conclusion that connectionist models are unable to learn a true torque rule is premature. The definition of torque as a product of number of weights and distance from the fulcrum is a rather simple multiplicative function, even if it has to be computed on both sides of the fulcrum and the larger torque selected as marking the descending side of the scale. An obvious, but untried way to see if networks can learn to compute and compare torques is to ensure that there are sufficient torque problems to the training set. Otherwise, the relatively few available torque problems could be easily swamped by the much larger number of problems that can correctly be solved by addition or even simpler rules involving weight or distance. This is particularly true when there are biases in the training set favoring equal-distance problems, thought to be necessary for capturing the stage-1 tendency to focus on weight information to the exclusion of distance information (McClelland, 1989; Shultz, et al., 1994).

Here we present simulations to test this hypothesis that prolonged training with sufficient numbers of torque problems would simulate learning to use torques. We apply several analyses of network responses to assess the use of torque and addition rules. We focus here on stages 3 and 4 and on the addition and torque rules thought to characterize those two stages, respectively (Boom, Hoijtink, & Kunnen, 2001; Jansen & van der Maas, 1997, 2002). We use newer sibling-descendant CC (SDCC) networks to study learning of a torque rule from prolonged exposure to torque problems.

A.2 Method

SDCC is an extension of the CC algorithm that allows a newly-recruited unit to be installed either on the current highest layer (as a sibling) or on its own higher layer (as a descendant) as in standard CC (Baluja & Fahlman, 1994). Sibling and descendant candidates compete with each other for recruitment. As in standard CC, the candidate whose activation correlates highest with network error during input phase is the one recruited. Because descendant candidates have extra, cascaded weights from the current highest layer of hidden units, they are typically penalized by having their correlations multiplied by .8. This tends to minimize network weights without harming generalization (Baluja & Fahlman, 1994). The basic idea of SDCC is to introduce more flexible network topologies and build whichever connectivity is most beneficial at the time of recruitment. Early simulations with SDCC have so far found that it creates smaller and more variable network topologies, but with the same functionality as standard CC (Shultz, 2006).

Because we focus here only on rules 3 and 4, we train SDCC networks only on conflict problems. For each of 20 networks, we randomly selected without replacement 40 addition problems and 40 torque problems for the training set. From the remainder of unselected problems, we randomly selected 12 addition problems and all of the remaining 12 torque problems for testing. After training to some epoch limit or to victory, whichever came first, we recorded the proportions correct of training problems and addition and torque test problems. Input and output coding and parameter settings were identical to the original CC simulation of balance-scale development (Shultz, et al., 1994). For each network, the recruitment pool contained eight sibling and eight descendant sigmoid units.

A.3 Results

We did some pilot testing to determine the epoch at which performance on the three measures reached asymptote. To explore this, we set last-epoch limits of 100 to 1000, in steps of 100. Figure A1 shows proportion correct on train, test-addition, and test-torque problems across these different levels of training. There were 20 SDCC networks in each run. It is apparent that proportion correct on all three sets of patterns peaked at about 800 epochs in these networks. The plots of performance on test problems suggest early success (across the first 100 epochs) on addition problems but not on torque problems which continue to improve up to about 800 epochs. The only region where the SD bars do not greatly overlap is at a 100-epoch limit. After

that, the different kinds of patterns (training, addition, and torque) do not differ significantly – the networks achieve a high level of success on all three.

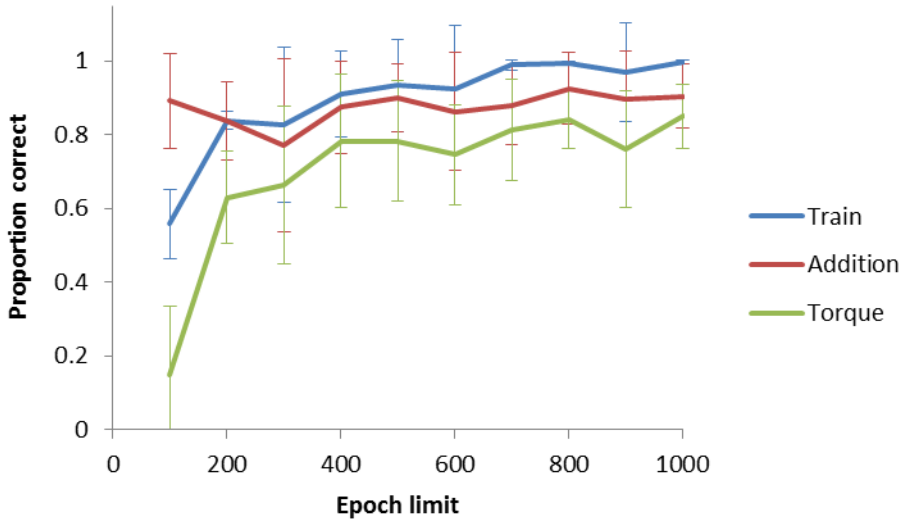

Figure A1. Mean proportion of patterns correct at various levels of training with SD bars.

To focus more precisely on the difference between following addition and torque rules, we compared networks trained up to a 100-epoch limit against those trained up to an 800-epoch limit. Mean numbers of recruited hidden units per network were 0.20 at the 100-epoch limit and 5.45 at the 800-epoch limit. Only 4 of the 20 networks run at the 100-epoch limit recruited any hidden units, and they each recruited a single hidden unit. These proportions were given an arcsine transformation in order to stabilize the variances (Hogg & Craig, 1995). These transformed values were subjected to a 2 x 3 mixed ANOVA in which epoch limit (100 vs. 800) served as a between-network factor and patterns (train, test-addition, and test-torque) served as a repeated-measures factor. There was a main effect of epoch limit, $F(1, 38) = 289$, $p < .0001$, a main effect of patterns, $F(2, 76) = 144$, $p < .0001$, and an interaction between them, $F(2, 76) = 80$, $p < .0001$. A follow-up repeated measures ANOVA of only test-addition problems showed no main effect of epoch limit, $F(1, 38) < 1$, indicating that networks mastered the addition problems during the first 100 epochs. Other simulations indicated that ordinary CC networks performed in a similar manner to these SDCC networks.

A.4 Discussion

It seems as though an addition rule can be followed without any (or many) hidden units, but a torque rule is more difficult thus requiring considerably more hidden units and epochs of training to recruit and consolidate the units. The fact that networks get 92% and 84% of addition and torque test problems, respectively, correct by 800 epochs suggests that they are following rule 4 by that point. In psychology experiments on the balance scale task, it has been conventional to considers 80% success on all problem types adequate for a diagnosis of rule 4 (Siegler, 1976).

These results show that constructive networks can learn to perform in conformity with an addition and a torque rule if given sufficient examples of each type of problem. Furthermore, these two rules emerge in the proper order, addition before torque. This order is natural for constructive networks that recruit hidden units only as needed to reduce error (Shultz, 2003). Indeed, such recruitments provide a computational explanation of this ordering in the sense that nonlinear hidden units underlie multiplication. The results also underscore that ANOVA applied to properly designed experiments can provide a valid diagnostic technique for rule-like performance.

A.5 References

Baluja, S., & Fahlman, S. E. (1994). Reducing network depth in the cascade-correlation learning architecture. Pittsburgh, PA: School of Computer Science, Carnegie Mellon University.

Boom, J., Hoijtink, H., & Kunnen, S. (2001). Rules in the balance: Classes, strategies, or rules for the balance scale task. *Cognitive Development, 16*, 717-735.

Hogg, R. V., & Craig, A. T. (1995). *Introduction to mathematical statistics* (5th ed.). Upper Saddle River, NJ: Prentice Hall.

Jansen, B. R. J., & van der Maas, H. L. J. (1997). Statistical test of the rule assessment methodology by latent class analysis. *Developmental Review, 17*, 321-357.

Jansen, B. R. J., & van der Maas, H. L. J. (2002). The development of children's rule use on the balance scale task. *Journal of Experimental Child Psychology, 81*, 383-416.

Langley, P. (1987). A general theory of discrimination learning. In D. Klahr, P. Langley & R. Neches (Eds.), *Production systems models of learning and development* (pp. 99-161). Cambridge, MA: MIT Press.

McClelland, J. L. (1989). Parallel distributed processing: Implications for cognition and development. In R. G. M. Morris (Ed.), *Parallel distributed processing: Implications for psychology and neurobiology* (pp. 8-45). Oxford, UK: Oxford University Press.

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

Quinlan, P. T., van der Maas, H. L. J., Jansen, B. R. J., Booij, O., & Rendell, M. (2007). Re-thinking stages of cognitive development: An appraisal of connectionist models of the balance scale task. *Cognition, 103*, 413-459.

Schapiro, A. C., & McClelland, J. L. (2009). A connectionist model of a continuous developmental transitionin the balance scale task. *Cognition, 110*, 395-411.

Schmidt, W. C., & Ling, C. X. (1996). A decision-tree model of balance scale development. *Machine Learning, 24*, 203-229.

Shultz, T. R. (2003). *Computational developmental psychology*. Cambridge, MA: MIT Press.

Shultz, T. R., Mareschal, D., & Schmidt, W. C. (1994). Modeling cognitive development on balance scale phenomena. *Machine Learning, 16*, 57-86.

Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology, 8*, 481-520.

Appendix B: Detecting Addition and Torque Rules with LCA and RAM in SDCC Networks

B.1 Introduction

In this appendix, we apply two different diagnostic techniques to SDCC networks to test their ability to transition to a genuine torque rule.

B.1.1 The RAM technique

The classic method of detecting rules on balance scale tasks is to test a participant with several problems of each of six types (Siegler, 1976). Three of these types are relatively simple problems with no conflict between weight and distance information: balance problems in which there are equal numbers of weights on each side of the scale placed equally distant from the fulcrum, weight problems in which one side has more weights than the other with the weights place equally distant from the fulcrum, and distance problems with the same number of weights on each side placed at different distances from the fulcrum. There are also three kinds of conflict problems in which one side of the scale has more weights and the other side has greater distance, making the prediction of which side will descend more difficult. Conflict-weight problems have greater torque on the side with more weight, conflict-distance problems have greater torque on the side with more distance, and conflict-balance problems have equal torques on each side of the fulcrum.

The classic Rule-assessment Method (RAM) examines the pattern of performance across these six problem types (Siegler & Chen, 2002). Use of rule 1 (weight information) is indicated by a pattern of correct performance on the balance, weight, and conflict-weight problems and incorrect performance on the distance, conflict-distance, and conflict-balance problems. Rule 2 (weight information, but use of distance when the weights are equal across the two sides) is characterized by the same pattern as rule 1, but with additionally correct performance on distance problems. In rule 3, weight and distance information are both used, yielding correct performance on the simple problems, but confusion on conflict problems.

Although Siegler (1976) suggested that rule 3 users guess on conflict problems, several later researchers have emphasized the use of other rules, particularly the addition rule, in which the side with the larger sum of weight and distance is predicted to descend (Boom, Hoijtink, & Kunnen, 2001; Ferretti, Butterfield, Cahn, & Kerkman, 1985; Jansen & van der Maas, 1997, 2002; Normandeau, Larivee, Roulin, & Longeot, 1989; Quinlan, van der Maas, Jansen, Booij, & Rendell, 2007). Rule 4 is characterized by successful performance on all six problem types, perhaps indicating use of the torque rule in which the side with the larger torque (weight x distance) is predicted to go down. To accommodate error variance in human performance, RAM users tolerate up to 20% deviant responses from these precise patterns of performance. Because these four rules tend to develop in order of their numerical designation, they are often taken as evidence that a child is in a particular stage of development.

B.1.2 The LCA technique

More recently, several researchers have argued that Latent Class Analysis (LCA) is a methodologically sounder way to detect rules (Boom, et al., 2001; Jansen & van der Maas, 1997, 2002; Quinlan, et al., 2007). In exploratory LCA, estimated parameters of a statistically-fitting model differ across latent classes, typically designating homogeneous groups of participants that differ from each other (McCutcheon, 1987). Individuals can be sorted into the latent classes based on membership probabilities estimated from the model.

In balance-scale research, the test problems typically come from the same types used by RAM researchers. But because LCA requires large numbers of participants and does better with small numbers of problems, non-diagnostic problems such as balance and weight problems are often omitted from the test set. Also, in this recent research, there is often a systematic attempt to distinguish the addition rule from the torque rule by including among the conflict test problems some that can solved by either addition or torque and others that can only be solved by torque.

B.1.3 The LCA vs. RAM debate

Advantages and disadvantages of these two methods for detecting balance-scale rules have been noted. RAM is favored for its transparency and ease of use with relatively small numbers of participants, convergence with other measures such as verbalization by the participants, stability over repeated measurements, prediction of which problems will best promote learning, and consistency across a wide variety of problems including conservation, fullness, shadow projection, and concepts of velocity, time, and distance (Siegler & Chen, 2002).

RAM has been criticized for using arbitrary scoring criteria (e.g., 20% tolerance), lack of statistical rigor, and inability to assess rules beyond those emphasized by the theoretical analysis of integrating two dimensions of information (Jansen & van der Maas, 2002). The criticized standard theoretical analysis involves a characterization of rule-based stages (Siegler, 1976). Children are assumed to start with one dimension, begin to include the other dimension when the first one fails to differentiate cases, eventually use both dimensions but become confused when these cues conflict, and finally integrate the two dimensions correctly. Although it is not clear how the addition rule could be derived from this stage analysis, it has been noted as a strategy by researchers using RAM (Ferretti, et al., 1985; Normandeau, et al., 1989).

LCA is favored for providing a statistical fit between a model and psychological data, avoiding arbitrary scoring criteria, allowing falsification of hypothesized rules, and discovery of new rules (Jansen & van der Maas, 1997, 2002). A counter argument is that only the issue of statistical fit uniquely favors LCA because RAM also allows for rule falsification and discovery, and choice of a significance level in LCA is no less arbitrary than a tolerance level in RAM (Siegler & Chen, 2002). LCA was further criticized for not providing stable assessments of rule use over short time periods and for requiring several orders of magnitude more subjects than RAM (Siegler & Chen, 2002). It is difficult to find an LCA study of the balance scale with fewer than about 500 participants.

These two diagnostic techniques each have their advantages and disadvantages and it is difficult to decide between them merely by applying them to (usually) different datasets, whether psychological data or computer simulations of psychological data. Here we apply both techniques to the same dataset, produced by a large number of SDCC networks.

B.2 Method

To approximate the number of cases used in recent LCA studies of human balance-scale development (Boom, et al., 2001), we trained 250 SDCC networks on randomly selected addition and torque problems for up to 100 epochs and 250 more SDCC networks for up to 800 epochs, as in the simulations of Appendix A. We replicated this five different times, for a total of 2500 networks. In each replication, we diagnosed rule following by both LCA and RAM methods. Our test patterns were four problems taken from a larger set of 15 problems used in recent LCA balance-scale studies of humans (Boom, et al., 2001), chosen because they distinguish addition from torque rules and possess a uniform absolute torque difference. It is now well known that problems with large absolute torque differences are easier for people and networks to solve at every stage (Ferretti & Butterfield, 1986; Ferretti, et al., 1985; Shultz, Mareschal, & Schmidt, 1994).

B.3 Results

For each of the five replications, the frequencies of the various response vectors were subjected to exploratory LCA using the LEM program (Vermunt, 1997), using default settings throughout. Model fit was evaluated with the Cressie-Read statistic which is essentially a generalization of the various chi-squared statistics (Cressie & Read, 1984). Following LCA conventions, we start with a 1-class model and increment classes by 1 until we obtain a non-significant Cressie-Read value (indicating that the model fits the data) or run out of degrees of freedom, whichever comes first. Because the sum of conditional probabilities across wrong and correct answers for each problem always sum to 1, we can summarize these parameters in a plot of estimated probabilities of correct responses (Figure B1). The plots for torque and addition rules follow their expected patterns – mostly correct on all four problems for the torque class and correct only on the addition problems for the addition class. The unknown class shows correct performance on one addition and one torque problem, middling performance on the other torque problem and poor performance on the other addition problem. This difficult-to-explain pattern did not replicate well.

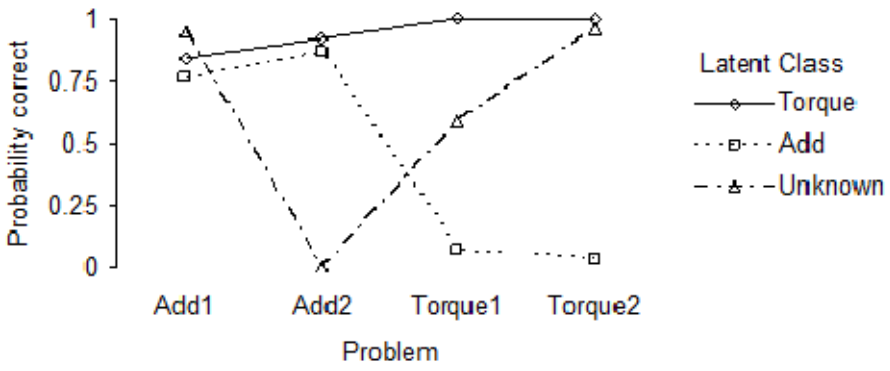

Figure B1. Conditional probabilities of SDCC networks being correct in replication 1.

To provide an indication of variability across replications, Figure B2 presents a similar plot for replication 2. All replications showed the expected patterns for torque and addition classes, but a highly variable pattern for the rare class we labeled as unknown. In replication 2, this class was characterized by middling performance on all four problems but slightly better performance on the addition problems than on the torque problems, and again did not replicate well.

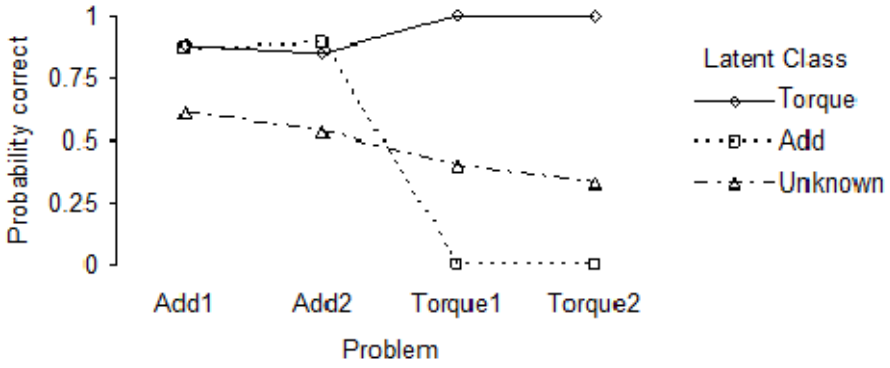

Figure B2. Conditional probabilities of SDCC networks being correct in replication 2.

Strict application of RAM to these data involved noting the frequencies of networks falling in the quintessential addition and torque patterns. These two cells always held the highest frequencies, containing 59% of the lightly-trained networks and 72% of the highly-trained networks. Allowing some Sieglerian latitude by also counting the frequencies of networks with only one of the two addition problems correct, these percentages rose to 82% and 94%, respectively.

B.4 Discussion

Confirming the results of the ANOVA technique for rule diagnosis in Appendix A, both LCA and RAM revealed an early addition rule and a later torque rule in SDCC networks trained on both addition and torque conflict problems. This further demonstrates that SDCC networks can learn a genuine torque rule from sufficiently informative examples and learn it later than a simpler addition rule, thus showing a natural progression from stage 3 to 4 on the balance scale. There was a strong tendency for LCA to also find a small third latent class that could not be reliably identified as representing any particular rule. The problem of high variability in LCA solutions is studied in greater detail with simulated data in Appendix C, where it is also shown that LCA cannot determine the correct number of classes.

B.5 References

Boom, J., Hoijtink, H., & Kunnen, S. (2001). Rules in the balance: Classes, strategies, or rules for the balance scale task. *Cognitive Development, 16*, 717-735.

Cressie, N., & Read, T. R. C. (1984). Multinomial goodness-of-fit tests. *Journal of the Royal Statistical Society B, 46*, 440-464.

Ferretti, R. P., & Butterfield, E. C. (1986). Are children's rule-assessment classifications invariant across instances of problem types? *Child Development, 57*, 1419-1428.

Ferretti, R. P., Butterfield, E. C., Cahn, A., & Kerkman, D. (1985). The classification of children's knowledge: Development on the balance-scale and inclined-plane tasks. *Journal of Experimental Child Psychology, 9*, 131-160.

Jansen, B. R. J., & van der Maas, H. L. J. (1997). Statistical test of the rule assessment methodology by latent class analysis. *Developmental Review, 17*, 321-357.

Jansen, B. R. J., & van der Maas, H. L. J. (2002). The development of children's rule use on the balance scale task. *Journal of Experimental Child Psychology, 81*, 383-416.

McCutcheon, A. L. (1987). *Latent class analysis*. Newbury Park, CA: Sage.

Normandeau, S., Larivee, S., Roulin, J. L., & Longeot, F. (1989). The balance-scale dilemma: Either the subject or the experimenter muddles through. *Journal of Genetic Psychology, 150*, 237-250.

Quinlan, P. T., van der Maas, H. L. J., Jansen, B. R. J., Booij, O., & Rendell, M. (2007). Re-thinking stages of cognitive development: An appraisal of connectionist models of the balance scale task. *Cognition, 103*, 413-459.

Shultz, T. R., Mareschal, D., & Schmidt, W. C. (1994). Modeling cognitive development on balance scale phenomena. *Machine Learning, 16*, 57-86.

Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology, 8*, 481-520.

Siegler, R. S., & Chen, Z. (2002). Development of rules and strategies: Balancing the old and the new. *Journal of Experimental Child Psychology, 81*, 446-457.

Vermunt, J. K. (1997). LEM: A general program for the analysis of categorical data. Tilburg University, Netherlands: Department of Methodology and Statistics.

Appendix C: LCA Yields Small and Unreliable Classes

C.1 Introduction

A debate has recently emerged regarding the application of Latent Class Analysis (LCA) to the diagnosis of rule-based stages in developmental psychology. Supporters favor LCA for providing a statistical fit between a model and psychological data, avoiding arbitrary scoring criteria, allowing falsification of hypothesized rules, and discovery of new rules (Jansen & van der Maas, 1997, 2002). Detractors counter that other techniques such as the Rule-assessment Method (RAM) also allow for rule falsification and rule discovery, and that choice of a significance level in LCA is no less arbitrary than a tolerance level in RAM (Siegler & Chen, 2002).

LCA was recently applied (Quinlan, van der Maas, Jansen, Booij, & Rendell, 2007) to a replication of an early computational model of children's development on the balance-scale task (Shultz, Mareschal, & Schmidt, 1994). Quinlan and colleagues (2007) criticized this computational model, which had used constructive neural networks, for not capturing these stages. In an invited response to the Quinlan et al. paper, we joined this debate, arguing among other points that LCA, as presently used, shows a strong tendency to identify small and unreliable classes (where each class represents a rule), and is thus suspect as method of validating stages in both psychological and modeling studies (Shultz & Takane, 2007). In their response to our rejoinder, our argument was dismissed because our demonstration used a 3-class model of 4 items, which yields an unidentified statistical model that does not converge during estimation (van der Maas, Quinlan, & Jansen, 2007). Here we present a 3-class model of 6 items, which yields an identified statistical model, to study the reliability problem in more detail, confirming that this problem is indeed severe enough to warrant extreme caution in using LCA to assess rule development. First, we present a brief recap of relevant background literature and issues.

C.1.1 LCA unreliability in psychological studies

The psychological literature on using LCA to assess balance-scale stages provides strong hints of unreliability in rule identification. Independent LCA studies of human balance-scale performance produce a number of small, leftover classes with mutually inconsistent interpretations. Boom and colleagues (Boom, Hoijtink, & Kunnen, 2001) found classes suggestive of Siegler's four rules, the addition rule, and several infrequent and uninterpretable classes. Jansen and van der Maas (1997) found classes for Siegler's first 2 rules, the addition rule, and a *no-balance* rule predicting that the scale would not balance, which was described as difficult to interpret. Jansen and van der Maas (2002) reported classes consistent with Siegler's four rules, addition, a *smallest-distance-down* rule, a *distance-and-guessing-when-weights-are-unequal* rule, a rule that seemed to combine Siegler's rule 3 with the addition rule, and additional difficult-to-interpret classes. Tellingly, it is the smaller classes that tend to be the most difficult to replicate across these human studies.

The problem with these small extra classes is not difficulty of interpretation. Humans often exhibit behavior that is difficult to interpret in terms of rules. The real problem with extra LCA classes is that they are small and inconsistent, suggesting that they might be random and meaningless.

Interesting in this context is Boom et al.'s (2001) distinction between classes and strategies. Classes are said to refer to a set of response patterns that are statistically similar, as revealed by say LCA. Strategies (or rules) refer to an interpreted procedure that could conceivably generate a statistical class. Classes that cannot be interpreted as being produced by sensible rules should not be treated as rules; they are merely statistical groupings that do not happen to fit a rule interpretation.

In summary, the only balance-scale rules to be reliably diagnosed by LCA in humans are Siegler's rules 1, 2, 4, and addition. Ignoring the small and difficult-to-interpret latent classes in Quinlan et al.'s (2007) study (Shultz & Takane, 2007), it is noteworthy that their LCA found evidence for Siegler's rules 1 and 2 and the addition rule in their replication of our original balance-scale simulation (Shultz et al., 1994). Apart from rule 4, these are precisely the same rules consistently found with children using LCA. Our accompanying manuscript and Appendices A and B provide clear evidence of neural networks also following a genuine torque rule (rule 4).

C.2 Method

To investigate the unreliability issue in LCA more deeply, we generated synthetic data from ideal addition and torque rules for six hypothetical conflict problems, three of which could be solved by either addition or torque comparisons and three of which could only be solved by comparison of torques. This transition from addition to torque rules is documented in our main paper and in Appendices A and B.

The simulation used class population sizes of .48 for a torque rule, .48 for an addition rule, and .04 for a small random class. There were 5000 replications of 500 cases each. There were six hypothetical balance-scale items, the first three of which could be solved by either an addition or a torque rule, and the remaining three of which could only be solved by a torque rule. For each of the ten replications, the frequencies of response patterns were subjected to exploratory LCA with the LEM program (Vermunt, 1997), using default parameter settings throughout.

C.3 Results

The mean conditional probabilities of being correct are plotted in Figure C1 for each item and latent class. As expected, the torque class was characterized by virtually perfect performance on each item, the addition class by virtually perfect performance on the first three items and failure on the last three items, and the random class by near chance performance.

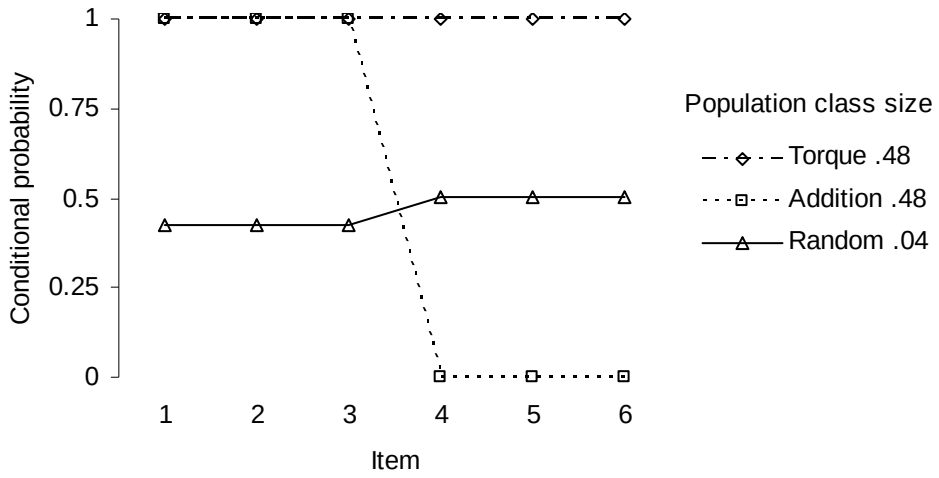

Figure C1. Mean conditional probabilities of being correct for three latent classes on six hypothetical items, the first three of which can be solved by either a torque rule or an addition rule and the last three of which can only be solved by a torque rule. Population class sizes are listed in the legend as proportions after the class name.

The 95% confidence bands for conditional probability estimates in the addition and random classes are plotted in Figure C2. They reveal far more variability in the random class than in the addition class. The torque class is excluded for clarity, but the confidence bands are as tight for that class as for the addition class, ranging between .9924 and 1.0.

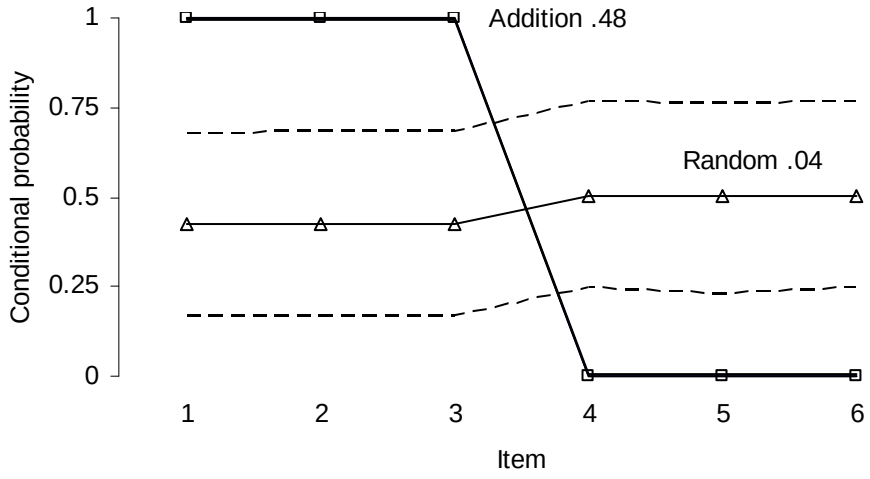

Figure C2. 95% confidence bands for the addition and random classes.

The standard deviations of these conditional probabilities are plotted in Figure C3, again for each item and class. The standard deviations for the small, random class are about 65 times greater than those for the large, systematic classes based on the torque and addition rules.

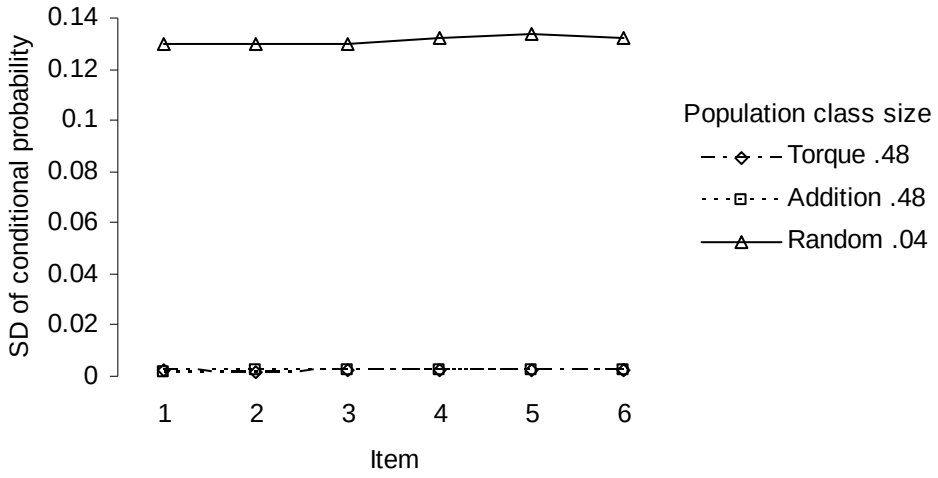

Figure C3. Standard deviations of the conditional probabilities in Figure C1.

These results approximately conform to the simplest binomial case. In the binomial distribution, the variance of the frequency *x* is *np*(1 - *p*), where *n* is sample size and *p* is *x*/*n*. From this, we can derive that $variance(p) = variance(x)/n^2 = p(1-p)/n$. If *p* is .5 (as in the random case) and *n* = 500 x .04 = 20, the SD of *p* is *sqrt*(.25 / 20) = 0.112. When *p* = 0 or 1 (as in the torque and addition cases), the SD is theoretically 0.

In further simulations, we found that roughly four times as many cases (*N* = 2000) were necessary to achieve the standard deviations that van der Maas et al. (2007) deemed to be acceptably low. Without any particular justification, they cited 0.035 as an acceptable overall mean standard deviation for conditional probabilities. We found a mean overall standard deviation of 0.022 for conditional probabilities with 2000 cases. The mean standard deviations for the small, random class was 70 times higher than the mean standard deviations for the two larger, systematic classes. Although 2000 participants may be feasible for computer simulations, it is unlikely that psychology researchers would run so many human participants in cognitive experiments, unless perhaps when running online experiments.

The relatively large variability for the small, random class is sufficient to produce many different patterns of performance across the six items, as shown in Figure C4, which plots conditional probabilities for the first five replications in an identical simulation, again with a more realistic 500 observations. In each replication, the pattern for the torque and addition classes is predictably regular – perfect performance for the torque class, and for the addition class perfect performance on the first three items and perfect failure on the last three items.

However, the pattern for the small, random class (solid line) is highly variable, affording many different interpretations across the replications.

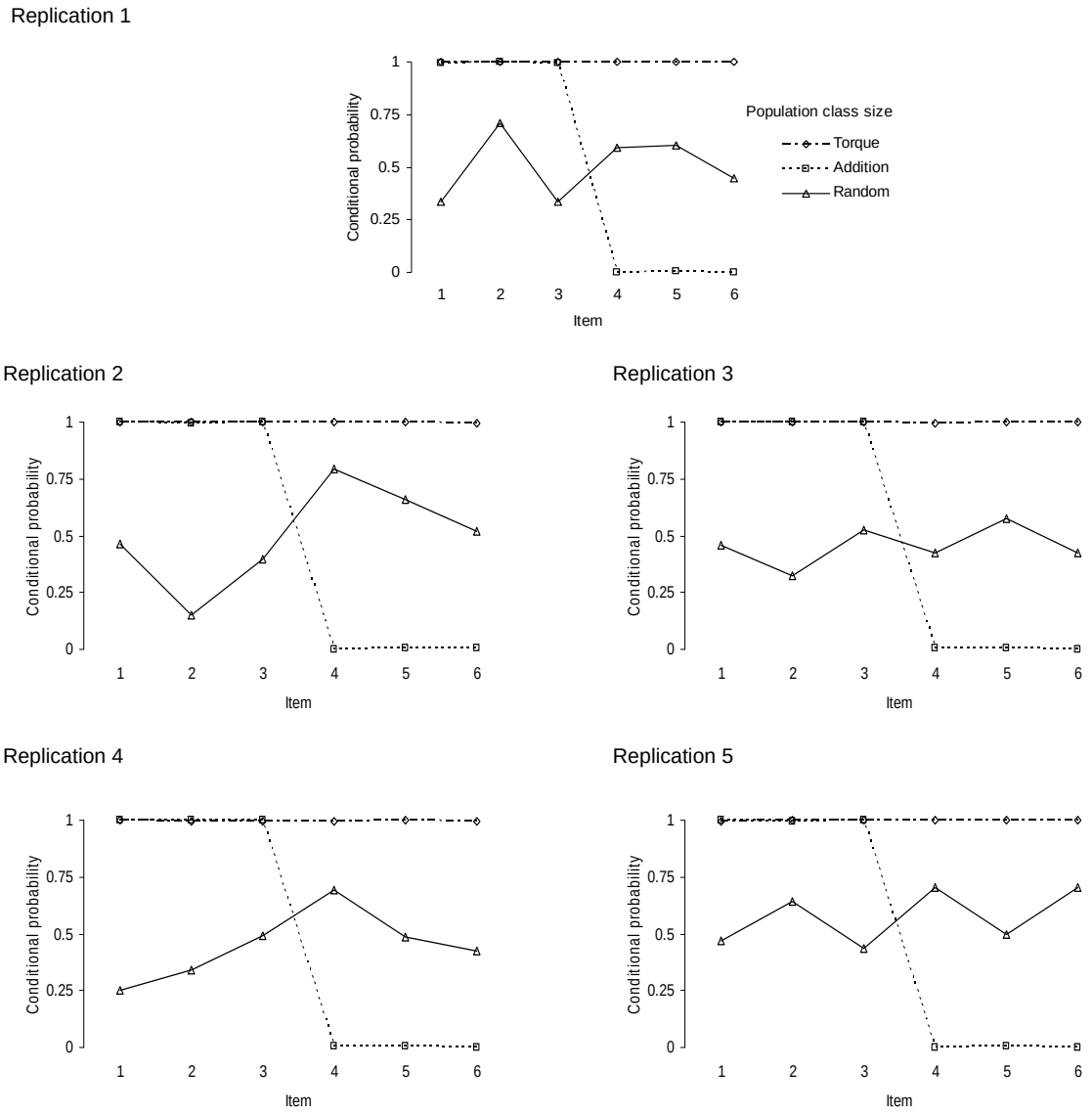

Figure C4. Estimated conditional probability of being correct for three distinct latent classes in five different replications.

The small, random class in replication 1 could be interpreted as success on item 2, failure on items 1 and 3, and guessing on items 4-6. The same class in replication 2 looks like success on items 4-6 and failure on items 1-3. Replication 3 features an alternating pattern with somewhat better performance on odd than even items, while replication 5 shows the reverse alternating pattern with better performance on evens than odds. In replication 4, there is a peak performance

at item 4. However, there is plenty of ambiguity in these interpretations. Such variability would be expected given the large standard deviations in conditional probabilities for the small, randomly-produced class.

C.3.1 Is Variability Caused by Smallness or Randomness?

An important question raised by these results is whether the variability of small, random latent classes is caused by the smallness of the class size or the random production of class members, or perhaps by both smallness and randomness. We studied this issue by switching the population size of the small, random class with that of one of the large systematic classes, the addition class. Thus, the torque and random classes each had a population size of .48, and the addition class had a population size of .04. Again, there were 5000 replications of 500 cases each.

The plot of mean conditional probability estimates over all 5000 replications in Figure C5 conforms to the expected pattern: for the torque class near perfect performance on all six items, for the addition class success only on the first three items and failure on the last three items, and for the random class random performance averaging to about .5.

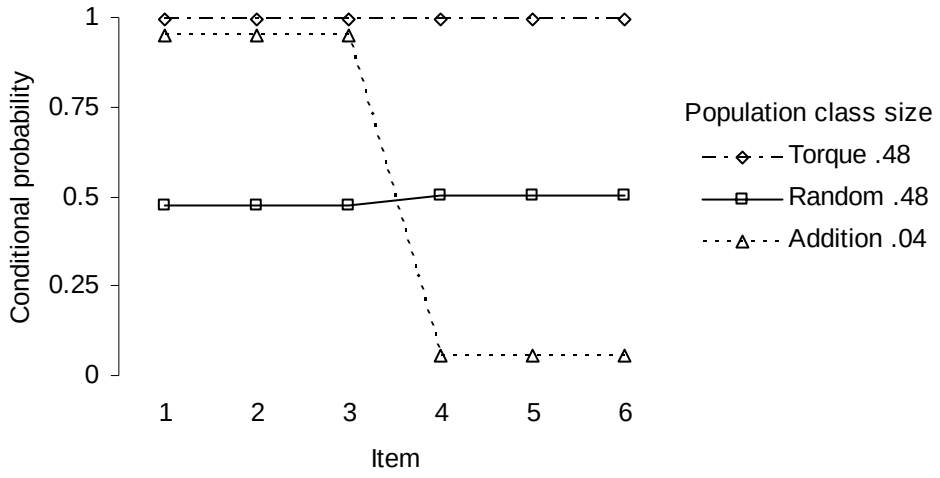

Figure C5. Mean conditional probabilities of being correct for three latent classes on six hypothetical items, the first three of which can be solved by either a torque rule or an addition rule and the last three of which can only be solved by a torque rule. Population class sizes are listed in the legend as proportions after the class name.

However, the plot of SDs of these same conditional probabilities in Figure C6 shows that variability is about 6.4 times greater in the large random condition than in the torque condition, and about 12.3 times greater in the small addition condition than in the torque condition.

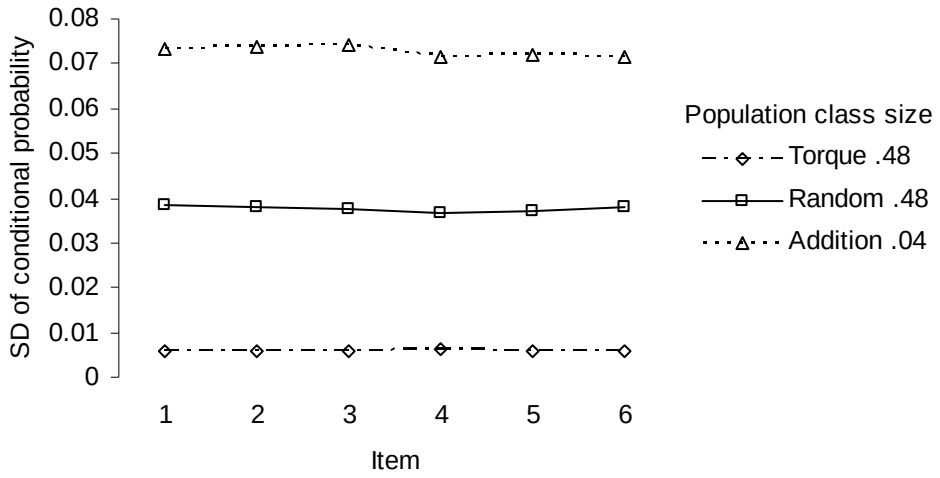

Figure C6. Standard deviations of the conditional probabilities in Figure 5.

C.3.2 Comparing across simulations

These results are summarized across the two foregoing simulations in Figure C7, which plots the mean SDs of conditional probabilities across items from Figures C3 (solid line) and C6 (dashed line). Examination of Figure C7 indicates that both smallness and randomness contribute to the variability of conditional probabilities. Within the constraints of the present parameter settings, smallness has a larger impact than does randomness in that the small but systematic condition is more variable than the large but random condition. The results suggest that even systematic classes will be difficult to replicate when they are small, and that random classes will be difficult to replicate even when they are large. If a class is both small and random, it will be especially unreliable.

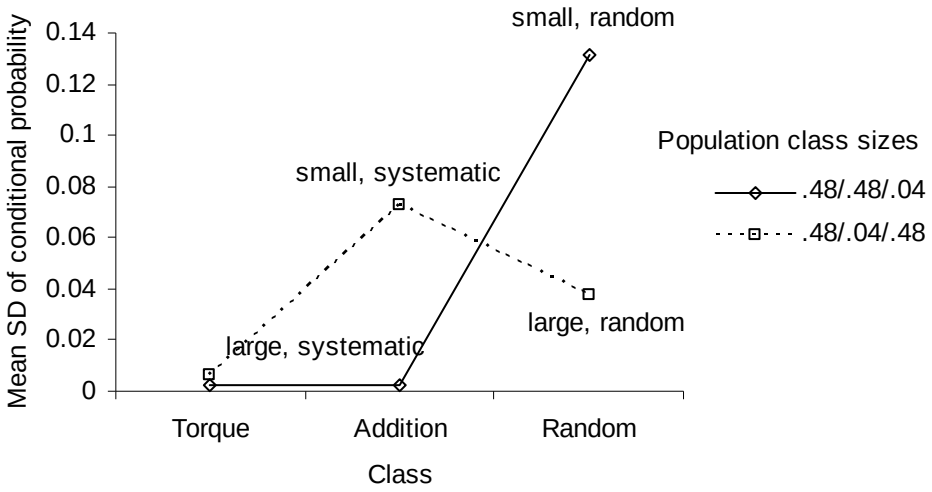

Figure C7. Mean SDs of conditional probabilities across items from Figures C3 and C6. Labels near the points describe the nature of the class in terms of its population size and randomness.

C.4 Discussion

Our present results, with an identified LCA model, confirm our previous conclusion with an unidentified model (Shultz & Takane, 2007) that LCA has serious problems in identifying small and unreliable rule classes. The dismissal of our previous demonstration for not having an identified model (van der Maas et al., 2007) proved to be a red herring. A deeper analysis here more precisely pinpointed and quantified the extent of this problem, finding that the standard deviations of conditional probabilities were 65-70 times greater for the small, random classes than for the large systematic classes. With such extraordinary variation, it is not surprising to also find that the pattern of these small, random classes do not replicate across different LCA replications. Of course, when the pattern across test items does not replicate, neither will its interpretation. Further investigation indicated that both class smallness and randomness contribute to increasing this variance, with smallness being more important under the current parameter settings.

Our critics argue that increased variation in small classes with larger samples is not a problem because the larger classes are unaffected by the additional smaller classes (van der Maas et al., 2007). We would agree that attention should be redirected from the unreliable, small classes to the reliable large classes, but it remains that our model was criticized for not covering the small, unreliable classes of often non-sensible rules (Quinlan et al., 2007).

Some of these problems with LCA can be traced to the LCA method and the frequency data used as input. The common method for parameter estimation in LCA is Maximum Likelihood Estimation (MLE), partly because MLE provides several statistical advantages. However, these advantages are present only when of the following conditions are all satisfied: 1) the fitted model

is correct, 2) the sample size is sufficiently large, and 3) other regularity conditions are met. We discuss each of these conditions and then an epistemological problem.

An LCA model consists of two parts, one statistical and the other parametric. The statistical part assumes independent trials and a multinomial probability distribution of multiple possible response patterns, only one of which occurs in each trial. The parametric part assumes several homogeneous groups in a heterogeneous population, with each group member responding to a set of items independently of other items (the Local Independence assumption – LI). Each group (represented as a latent class) is characterized by its size and a set of conditional probabilities of responses to particular items. To satisfy the LI assumption, a large number of latent classes typically must be assumed, but this tends to produce latent classes that are difficult to interpret (Bartholomew, 1987; Hagenaars, 1990; Qu, Tan, & Kutner, 1996).

The benefits of MLE emerge only with a sufficiently large sample. However, what constitutes a large sample is controversial (Hagenaars, 1990; Wickens, 1989). There are 2^n possible response patterns when there are n dichotomous items, and reliably estimating the probabilities of these response patterns requires a large number of participants. There should be at least one, but preferably five or more cases in each response pattern. This condition can be difficult to satisfy, particularly when some response patterns rarely occur, which happens as the number of response patterns increases. Although this problem is under active consideration (Bartholomew & Leung, 2002; Hoijtink, 1998; Reiser & Lin, 1999), there is currently no commonly-accepted solution.

One of the regularity conditions for the standard asymptotic properties of MLE is that LCA parameters must reside in the interior of the parameter space. However, it is often the case that important parameter values (like those representing crisp rules) are actually on the boundaries of the parameter space with conditional probabilities of 0 or 1, as exemplified throughout LCA analyses of balance-scale results. Although there are some attempts to extend asymptotic theory to cover cases in which estimates are subject to inequality constraints (Dijkstra, 1992; Shapiro, 1985, 1988), this complicates the theory enough to prevent integration of these efforts into LCA literature and software. The only known practical solutions are resampling methods, such as the parametric bootstrap (Aitkin, Anderson, & Hinde, 1981; Langeheine, Pannekoek, & van de Pol, 1996).

Moreover, even when all three conditions are met, there is an unresolved epistemological issue: there is no statistical method to determine the correct number of latent classes. Although one might argue that goodness-of-fit tests can determine the number of significant latent classes, such tests are not designed for this – they are instead designed to determine how many latent classes are needed to satisfy the LI assumption. The number of latent classes naturally increases with sample size because with a large sample even a small departure of the model from the data becomes significant, and in order to get a satisfactory fit, the number of latent classes has to be increased. With number of latent classes directly dependent on sample size, there is no correct number of latent classes in LCA. The number of latent classes to extract can also depend on the

purpose of the analysis. There are typically some researchers wanting most of the variability in the data to be explained by a model and some who are satisfied with less. The former will retain all the latent classes, while the latter retain only the most frequent classes.

These are the same reasons that a statistical approach to factor analysis, a related technique for finding latent structure, has never found the correct number of common factors defining human intelligence and other characteristics. With a large sample size, even small correlations become significantly different from zero, and a large number of factors are required to explain the correlations. Researchers seeking a simpler, more unified picture of intelligence can use smaller samples, whereas those convinced of the complexity of intelligence can support their position with larger samples. Unless and until such statistical problems are resolved, our recommendation is to confine interpretation of latent classes to those that are replicated across studies. In particular, computational modelers should not bother chasing all of the latent classes found in LCA of children's responses, as some researchers have mandated (Quinlan et al., 2007).

C.5 References

Aitkin, M., Anderson, D., & Hinde, J. (1981). Statistical modelling of data on teaching styles. *Journal of the Royal Statistical Society B, Series A, 144*, 419-461.

Bartholomew, D. J. (1987). *Latent variable models and factor analysis*. London: Charles Griffin and Company.

Bartholomew, D. J., & Leung, S. O. (2002). A goodness of fit test for sparse 2^p contingency tables. *British Journal of Mathematical and Statistical Psychology, 55*, 1-15.

Boom, J., Hoijtink, H., & Kunnen, S. (2001). Rules in the balance: Classes, strategies, or rules for the balance scale task. *Cognitive Development, 16*, 717-735.

Dijkstra, T. K. (1992). On statistical inference with parameter estimates on the boundary of the parameter space. *British Journal of Mathematical and Statistical Psychology, 45*, 289-309.

Hagenaars, J. A. (1990). *Categorical longitudinal data*. Newbury Park, CA: Sage.

Hoijtink, H. (1998). Constrained latent class analysis using Gibbs sampler and posterior predictive p-values: Applications to educational testing. *Statistics Sinica, 8*, 691-711.

Jansen, B. R. J., & van der Maas, H. L. J. (1997). Statistical test of the rule assessment methodology by latent class analysis. *Developmental Review, 17*, 321-357.

Jansen, B. R. J., & van der Maas, H. L. J. (2002). The development of children's rule use on the balance scale task. *Journal of Experimental Child Psychology, 81*, 383-416.

Langeheine, R., Pannekoek, J., & van de Pol, F. (1996). Bootstrap goodness-of-fit measures in categorical data analysis. *Sociological Methods and Research, 24*, 493-516.

Qu, Y., Tan, M., & Kutner, M. H. (1996). Random effects model in latent class analysis for evaluating accuracy of diagnostic tests. *Biometrics, 52*, 797-810.

Quinlan, P. T., van der Maas, H. L. J., Jansen, B. R. J., Booij, O., & Rendell, M. (2007). Re-thinking stages of cognitive development: An appraisal of connectionist models of the balance scale task. *Cognition, 103*, 413-459.

Reiser, M., & Lin, Y. (1999). A goodness-of-fit test for the latent class model when expected frequencies are small. In M. Sobel & M. Decker (Eds.), *Sociological methodology* (pp. 81-111). Boston: Blackwell.

Shapiro, A. (1985). Asymptotic distribution of test statistics in the analysis of moment structures under inequality constraints. *Biometrika, 72,* 133-144.

Shapiro, A. (1988). Toward a unified theory of inequality constrained testing in multivariate analysis. *International Statistical Review, 56,* 49-62.

Shultz, T. R., Mareschal, D., & Schmidt, W. C. (1994). Modeling cognitive development on balance scale phenomena. *Machine Learning, 16,* 57-86.

Shultz, T. R., & Takane, Y. (2007). Rule following and rule use in simulations of the balance-scale task. *Cognition, 103,* 460-472.

Siegler, R. S., & Chen, Z. (2002). Development of rules and strategies: Balancing the old and the new. *Journal of Experimental Child Psychology, 81,* 446-457.

van der Maas, H. L. J., Quinlan, P. T., & Jansen, B. R. J. (2007). Towards better computational models of the balance scale task: A reply to Shultz and Takane. *Cognition, 103,* 473-479.

Vermunt, J. K. (1997). LEM: A general program for the analysis of categorical data. Tilburg University, Netherlands: Department of Methodology and Statistics.

Wickens, T. D. (1989). *Multiway contingency tables analysis for the social sciences*. Hillsdale, NJ: Erlbaum.

References

Baetu, I., & Shultz, T. R. (2010). Development of prototype abstraction and exemplar memorization. In S. Ohlsson & R. Catrambone (Eds.), *Proceedings of the 32nd Annual Conference of the Cognitive Science Society* (pp. 814-819). Austin, TX: Cognitive Science Society.

Baluja, S., & Fahlman, S. E. (1994). Reducing network depth in the cascade-correlation learning architecture. Pittsburgh, PA: School of Computer Science, Carnegie Mellon University.

Berthiaume, V. G., Onishi, K. H., & Shultz, T. R. (2008). A computational developmental model of the implicit false belief task. In B. C. Love, K. McRae & V. M. Sloutsky (Eds.), *Proceedings of the 30th Annual Conference of the Cognitive Science Society* (pp. 825-830). Austin, TX: Cognitive Science Society.

Boom, J., Hoijtink, H., & Kunnen, S. (2001). Rules in the balance: Classes, strategies, or rules for the balance scale task. *Cognitive Development, 16*, 717-735.

Buckingham, D., & Shultz, T. R. (2000). The developmental course of distance, time, and velocity concepts: A generative connectionist model. *Journal of Cognition and Development, 1*, 305-345.

Dandurand, F., Berthiaume, V., & Shultz, T. R. (2007). A systematic comparison of flat and standard cascade-correlation using a student-teacher network approximation task. *Connection Science, 19*, 223-244.

Dandurand, F., & Shultz, T. R. (2009). Modeling acquisition of a torque rule on the balance-scale task. In N. A. Taatgen & H. v. Rijn (Eds.), *Proceedings of the 31st Annual Conference of the Cognitive Science Society* (pp. 1541-1546). Austin, TX: Cognitive Science Society.

Dandurand, F., & Shultz, T. R. (2010). Automatic detection and quantification of growth spurts. *Behavior Research Methods, 42*(3), 809-823. doi: 10.3758/BRM.42.3.809

Durbin, R., & Rumelhart, D. E. (1989). Product units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation, 1*, 133–142.

Evans, J. B. T. (2003). In two minds: dual-process accounts of reasoning. *Trends in Cognitive Sciences, 7*(10), 454-459.

Evans, J. B. T. (2006). The heuristic-analytic theory of reasoning: Extension and evaluation. *Psychonomic Bulletin & Review, 13*(3), 378-395.

Evans, J. B. T. (2010). *Thinking twice: Two minds in one brain*. Oxford: Oxford University Press.

Evans, J. B. T., & Wason, P. C. (1976). Rationalization in a reasoning task. *British Journal of Psychology, 67*, 479-486.

Evans, V. C., Berthiaume, V. G., & Shultz, T. R. (2010). Toddlers' transitions on non-verbal false-belief tasks involving a novel location: a constructivist connectionist model

Proceedings of the Ninth IEEE International Conference on Development and Learning (pp. 225-230). Ann Arbor, MI: IEEE.

Fahlman, S. E., & Lebiere, C. (1990). The cascade-correlation learning architecture. In D. S. Touretzky (Ed.), *Advances in neural information processing systems 2* (pp. 524-532). Los Altos, CA: Morgan Kaufmann.

Ferretti, R. P., & Butterfield, E. C. (1986). Are children's rule-assessment classifications invariant across instances of problem types? *Child Development, 57*, 1419-1428.

Ferretti, R. P., Butterfield, E. C., Cahn, A., & Kerkman, D. (1985). The classification of children's knowledge: Development on the balance-scale and inclined-plane tasks. *Journal of Experimental Child Psychology, 9*, 131-160.

Frank, M. J., Cohen, M., & Sanfey, A. G. (2009). Multiple systems in decision making. *Current Directions in Psychological Science, 18*, 73-77.

Hahn, U., & Nakisa, R. (2000). German inflection: single route or dual route? *Cognitive Psychology, 41*, 313-360.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780.

Jansen, B. R. J., & van der Maas, H. L. J. (1997). Statistical test of the rule assessment methodology by latent class analysis. *Developmental Review, 17*, 321-357.

Jansen, B. R. J., & van der Maas, H. L. J. (2002). The development of children's rule use on the balance scale task. *Journal of Experimental Child Psychology, 81*, 383-416.

Kahneman, D. (2011). *Thinking, fast and slow*. New York: Farrar, Straus and Giroux.

Kahneman, D., & Frederick, S. (2005). A model of heuristic judgment. In K. J. Holyoak & R. G. Morrison (Eds.), *The Cambridge handbook of thinking and reasoning* (pp. 267-293). New York: Cambridge University Press.

Mareschal, D., & Shultz, T. R. (1999). Development of children's seriation: A connectionist approach. *Connection Science, 11*, 149-186.

McClelland, J. L. (1989). Parallel distributed processing: Implications for cognition and development. In R. G. M. Morris (Ed.), *Parallel distributed processing: Implications for psychology and neurobiology* (pp. 8-45). Oxford, UK: Oxford University Press.

Normandeau, S., Larivee, S., Roulin, J. L., & Longeot, F. (1989). The balance-scale dilemma: Either the subject or the experimenter muddles through. *Journal of Genetic Psychology, 150*, 237-250.

Oshima-Takane, Y., Takane, Y., & Shultz, T. R. (1999). The learning of first and second pronouns in English: Network models and analysis. *Journal of Child Language, 26*, 545-575.

Posner, M. I., & Snyder, C. R. R. (1975). Attention and cognitive control. In R. L. Solso (Ed.), *Information processing and cognition: The Loyola Symposium* (pp. 55-85). New York: Wiley.

Quinlan, P. T., van der Maas, H. L. J., Jansen, B. R. J., Booij, O., & Rendell, M. (2007). Re-thinking stages of cognitive development: An appraisal of connectionist models of the balance scale task. *Cognition, 103*, 413-459.

Reber, A. (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General, 118*(3), 219-235.

Reber, A., & Lewis, S. (1977). Implicit learning: an analysis of the form and structure of a body of tacit knowledge. *Cognition, 5*, 333-361.

Schapiro, A. C., & McClelland, J. L. (2009). A connectionist model of a continuous developmental transitionin the balance scale task. *Cognition, 110*, 395-411.

Schmidt, W. C., & Ling, C. X. (1996). A decision-tree model of balance scale development. *Machine Learning, 24*, 203-229.

Shepard, R. N. (2008). The step to rationality: The efficacy of thought experiments in science, ethics, and free will. *Cognitive Science, 32*, 3-35. doi: 10.1080/03640210701801917

Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. *Psychological Review, 84*, 127-190.

Shultz, T. R. (1998). A computational analysis of conservation. *Developmental Science, 1*, 103-126.

Shultz, T. R. (2003). *Computational developmental psychology*. Cambridge, MA: MIT Press.

Shultz, T. R. (2006). Constructive learning in the modeling of psychological development. In Y. Munakata & M. H. Johnson (Eds.), *Processes of change in brain and cognitive development: Attention and performance XXI.* (pp. 61-86). Oxford, UK: Oxford University Press.

Shultz, T. R. (2011). Computational modeling of infant concept learning: The developmental shift from features to correlations. In L. M. Oakes, C. H. Cashon, M. Casasola & D. H. Rakison (Eds.), *Infant perception and cognition: Recent advances, emerging theories, and future directions* (pp. 125-152). New York: Oxford University Press.

Shultz, T. R., & Bale, A. C. (2001). Neural network simulation of infant familiarization to artificial sentences: Rule-like behavior without explicit rules and variables. *Infancy, 2*, 501-536.

Shultz, T. R., & Bale, A. C. (2006). Neural networks discover a near-identity relation to distinguish simple syntactic forms. *Minds and Machines, 16*, 107-139.

Shultz, T. R., Berthiaume, V. G., & Dandurand, F. (2010). Bootstrapping syntax from morpho-phonology *Proceedings of the Ninth IEEE International Conference on Development and Learning* (pp. 52-57). Ann Arbor, MI: IEEE.

Shultz, T. R., Buckingham, D., & Oshima-Takane, Y. (1994). A connectionist model of the learning of personal pronouns in English. In S. J. Hanson, T. Petsche, M. Kearns & R. L. Rivest (Eds.), *Computational learning theory and natural learning systems, Vol. 2: Intersection between theory and experiment* (pp. 347-362). Cambridge, MA: MIT Press.

Shultz, T. R., & Cohen, L. B. (2004). Modeling age differences in infant category learning. *Infancy, 5*, 153-171.

Shultz, T. R., & Fahlman, S. E. (2010). Cascade-Correlation. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning, Part 4/C* (pp. 139-147). Heidelberg, Germany: Springer-Verlag.

Shultz, T. R., & Gerken, L. A. (2005). A model of infant learning of word stress. *Proceedings of the Twenty-seventh Annual Conference of the Cognitive Science Society* (pp. 2015-2020). Mahwah, NJ: Erlbaum.

Shultz, T. R., Mareschal, D., & Schmidt, W. C. (1994). Modeling cognitive development on balance scale phenomena. *Machine Learning, 16*, 57-86.

Shultz, T. R., & Rivest, F. (2001). Knowledge-based cascade-correlation: Using knowledge to speed learning. *Connection Science, 13*, 1-30.

Shultz, T. R., Rivest, F., Egri, L., Thivierge, J.-P., & Dandurand, F. (2007). Could knowledge-based neural learning be useful in developmental robotics? The case of KBCC. *International Journal of Humanoid Robotics, 4*, 245-279.

Shultz, T. R., & Takane, Y. (2007). Rule following and rule use in simulations of the balance-scale task. *Cognition, 103*, 460-472.

Shultz, T. R., Thivierge, J. P., & Laurin, K. (2008). Acquisition of concepts with characteristic and defining features. In B. C. Love, K. McRae & V. M. Sloutsky (Eds.), *Proceedings of the 30th Annual Conference of the Cognitive Science Society* (pp. 531-536). Austin, TX: Cognitive Science Society.

Shultz, T. R., & Vogel, A. (2004). A connectionist model of the development of transitivity *Proceedings of the Twenty-sixth Annual Conference of the Cognitive Science Society* (pp. 1243-1248). Mahwah, NJ: Erlbaum.

Siegler, R. S. (1976). Three aspects of cognitive development. *Cognitive Psychology, 8*, 481-520.

Sirois, S., & Shultz, T. R. (1998). Neural network modeling of developmental effects in discrimination shifts. *Journal of Experimental Child Psychology, 71*, 235-274.

Sjogaard, S. (1992). Generalization in cascade-correlation networks *Neural Networks for Signal Processing: Proceedings of the 1992 IEEE-SP Workshop* (pp. 59-68).

Sloman, S. A. (1996). The empirical case for two systems of reasoning. *Psychological Bulletin, 119*, 3-22.

Stanovich, K. E. (2012). On the distinction between rationality and intelligence: Implications for understanding individual differences in reasoning. In K. Holyoak & R. Morrison (Eds.), *The Oxford handbook of thinking and reasoning* (pp. 343-365). New York: Oxford University Press.

Sun, R., Slusarz, P., & Terry, C. (2005). The interaction of the explicit and the implicit in skill learning: A dual-process approach. *Psychological Review, 112*, 159-192.

Thompson, V. A., Prowse Turner, J. A., & Pennycook, G. (2011). Intuition, reason, and metacognition. *Cognitive Psychology, 63*, 107-140.

van der Maas, H. L. J., & Jansen, B. R. J. (2003). What response times tell of children's behavior on the balance scale task. *Journal of Experimental Child Psychology, 85*, 141-177. doi: 10.1016/S0022-0965(03)00058-4

van Rijn, H., van Someren, M., & van der Maas, H. (2003). Modeling developmental transitions on the balance scale task. *Cognitive Science, 27*, 227-257.

Wason, P. C., & Evans, J. B. T. (1975). Dual processes in reasoning? *Cognition, 3*, 141-154.

Wilson, T. D. (2002). *Strangers to ourselves*. Cambridge, MA: Harvard University Press.