Progressive Chunking for Motor Enhancement

Lia Veith

First Supervisor: Dr. Russel W. Chan
Second Supervisor: Prof. Dr. Ing. Willem B. Verwey

Department of Cognitive Psychology and Ergonomics
Faculty of Behavioural, Management and Social Sciences

University of Twente

July 5th, 2022

Author Note

The data which this thesis is based on has been collected in collaboration with two other BSc psychology students, namely Rian Telman and Nerijus Bertalis.

Abstract

Motor learning is an essential skill needed in everyday life. However, it seems that it is still debated what the best strategy entails when learning a motor task. There is consensus that learning tasks that are complex and serial in nature may benefit from progressive learning, which is also supported by the Cognitive Framework for Sequential Motor Behaviour (C-SMB). The framework details that singular movements are chunked together to ensure faster and more efficient performance. In order to examine whether this is indeed the case, participants (N = 24) performed a Dance-Step Discrete Sequence Production (DS-DSP) Task, either learning the sequences as a whole (N = 12) or progressively (N = 12). There seemed to be no actual difference between the two groups once learning was completed. However, significant impact on the process of learning was found, both in terms of accuracy and response time, pointing towards differences in cognitive processing between the groups. This was further confirmed when looking at the response time pattern of the single steps taken. The progressive learners showed significantly faster processing in the first few steps of the sequence, while their later steps followed a similar pattern to the whole learners. It is possible that whole learners used an associative mode throughout the full sequence in order to execute the steps. The results allude that progressive learners used two different strategies across different parts of the sequence, namely processing its first part in chunking mode, and the latter steps in association mode. This points to chunking only being activated under specific circumstances when performing a DS-DSP task.

Keywords: Motor Learning, Motor Chunking, Discrete Sequence Production Task

Table of Contents

1. Introduction

In our daily lives, motor skills provide an essential basis to our regular functioning. We need to know a vast variety of movements which are acquired throughout the years. Motor skill encompasses several activities, from tasks such as typewriting and piano playing, which require smaller and localized movements, to tasks such as swimming and climbing, requiring full body engagement. Those motor skills involve various combinations of cognitive, perceptual, and motor processes to different degrees (Adams, 1987).

In order to perfect these skills, learning is crucial (Adams, 1987). Motor sequence learning (MSL) defines the process by which – thanks to repeated practice – one is able to combine multiple movements to be performed as one in terms of movement efficiency (Doyon, 2008; Doyon et al., 2018). This type of learning is known to go through different phases of slower learning, faster learning, and automaticity (Doyon et al., 2018; Verwey et al., 2015). If learning is successful, memory consolidation occurs, which means that the learned motor sequence has been embedded in the structure of the brain and can be accessed without issue (Doyon, 2008).

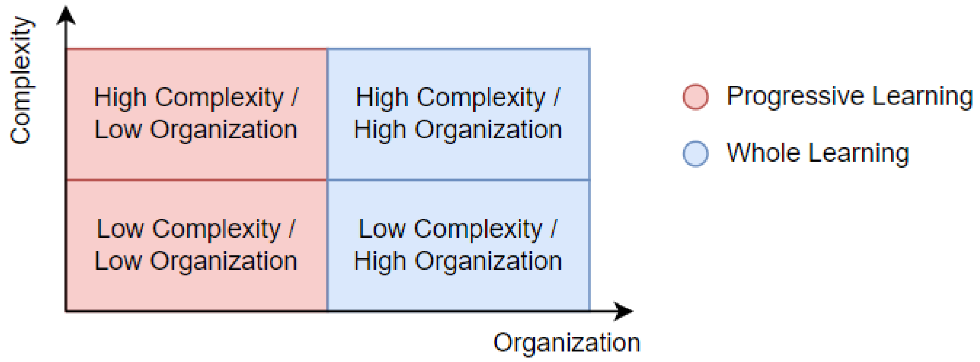
Over the years, the best way to learn has been debated from multiple angles. The central concern of this thesis is the discussion on whether whole or partial learning is better when practicing a motor learning task (Naylor & Briggs, 1963; Schmidt & Wrisberg, 2008). Alternative deliberations involve using blocked or variable practice (Wu et al., 2011; Wulf & Schmidt, 1997), or facilitating the advantages of sleep between multiple practices (Doyon, 2008; Walker et al., 2003). However, whilst the cognitive underpinnings of those methods have been fairly established, the debate of whole versus part practice and its mental processes is unclear. Although the effects of whole and part practice have been outlined in sports (Lingyun, 2021), the exact cognitive underpinnings have not been clearly investigated and therefore form the main focus of the current thesis.

1.1. The Role of Task Parameters in Methods of Learning

In order to optimize learning in general, there are multiple task parameters which can facilitate or inhibit training. Overall, effectiveness of learning boils down to the difficulty of the task to-be learned. The difficulty is made up of task complexity and task organization, which are both determined by the relative performance levels of those domains by learners (Naylor & Briggs, 1963). Hereby, the parameter 'task complexity' is defined as "the demands placed on [the learner's] information-processing and/or memory-storage capacities by each of the task dimensions independently" (Naylor & Briggs, 1963, p. 217). Subsequently, the parameter 'task organization' is explained as "the degree to which separate task dimensions have meaningful relationships with each other" (Naylor & Briggs, 1963, p. 217). As an example for a high complexity task with low organization, one can consider the execution of an aerobics routine; there are many movements which require a person to activate a large amount of mental capacity. For example, each step is independent from the next one, therefore not requiring high organization. Opposite of that would be a task such as shooting an arrow; the complexity is fairly low; there is only one motion to consider. However, this task requires high levels of organization: your posture, your grip on the bow, the way you pull the arrow back; all parts of the body must work together in order to perform well and hit the target.

Figure 1

Different Modes of Learning depending on Task Complexity and Task Organization



Note. Progressive learning works best in tasks with low organization (as seen in red). Whole learning works best in tasks with high organization (as seen in blue). Neither depends on level of task complexity. Visualization based on descriptions in "Effects of task complexity and task organization on the relative efficiency of part and whole training methods" by J. C. Naylor and G. E. Briggs, 1968, *Journal of Experimental Psychology, 65*(3), (https://doi.org/10.1037/h0041060).

Depending on the nature of the task, the ideal way of learning may differ (see Figure 1). Generally speaking, it is possible to practice a task as a whole or in progressive parts. In this case, whole practice refers to practicing a task in its entirety from the start and trying to follow it to the best of your ability from beginning to end. If a task is high in organization, it was found more effective to practice it as a whole (Naylor & Briggs, 1963). This is the case as tasks high in organization usually require the trainee to have high levels of integration between the different types of movement. It is necessary to practice as a whole, since the singular aspects all depend on each other (Naylor & Briggs, 1963).

In contrast, practicing in progressing parts involves breaking down a task into multiple parts and introducing more and more elements throughout practice step by step. Learners performing low organization tasks were significantly better when using a progressing parts approach to training (Naylor & Briggs, 1963). This result is observed since the task requires little interrelation between its singular aspects, making part practice more sensible (Naylor & Briggs, 1963).

Schmidt and Wrisberg (2008) suggested a slightly different but still related approach; tasks with a clearly defined beginning and end should be practiced as a whole, regardless of complexity and organization (Schmidt & Wrisberg, 2008). However, serial tasks with no clearly defined beginning or end point should follow the same approach that Naylor and Briggs (1968) have suggested; tasks with high organization should use whole practice as the preferred method, while tasks with low organization should be practiced in parts (Schmidt & Wrisberg, 2008). Importantly, there are multiple approaches to part practice.

1.2. Whole Versus Part Practice

There are several ways to segment a sequence that fall under the umbrella of part practice. In the part-whole method the task is practiced in separate smaller parts, and these are combined at the end of learning. Also, it is possible to practice in different parts, but also include the progression between the two adjacent parts in the practice. Lastly, in the progressive part practice, the task is progressively extended as to include more and more elements.

Fontana et al. (2009) conducted a meta-analysis in order to examine the theories of Naylor and Briggs (1968) and Schmidt and Wrisberg (2008) regarding the effectiveness of whole versus part practice including various types of partial learning. According to this analysis, neither Naylor and Briggs's (1968) nor Schmidt and Wrisberg's (2008) assumptions regarding whole versus part practice have been confirmed by statistical analyses of the studies examined (Fontana et al., 2009). Overall, this study considered part-whole practice, progressive practice, whole practice, and different chaining strategies. However, no difference was found between the various modes of segmentation and types of learning tasks (Fontana et al., 2009). This conclusion in line with a different study in which participants practiced a four part motor learning sequence by pressing on three different buttons (Hansen & Tremblay, 2005). The study has found that practicing the sequence as a whole, in two separate parts, or with overlap of parts led to no difference in regard to improvement of performance. Rather, all three conditions were able to yield relatively long-term improvement of performance of the movement sequence (Hansen & Tremblay, 2005). Importantly, partial practice thereby leads to equal improvement of learning, pointing towards different types of optimizations based on the mode of learning.

On the other hand, there is the argument that Naylor and Briggs (1968) were indeed able to prove their hypothesis with their own experiment, albeit it is not related to motor learning. During their study, participants were asked to predict stimuli based on type of stimulus, placement of stimulus, and number of stimuli. Different stimuli where hereby associated with different patterns of the predictors. Associations differed both in task organization and complexity. Additionally, Fontana et al. (2009) still found that mean effect sizes indicate support for Naylor and Briggs' hypotheses; the relationship between organization and complexity of a task requiring different types of learning was supported. In conclusion, low-organization and high-complexity tasks indeed performed best with part practice while high-organization and low-complexity tasks were learned best with whole practice (Fontana et al., 2009). Furthermore, a similar pattern was found by Schmidt and Wrisberg (2008): mean effect sizes showed that serial tasks with low organization were better-learned using part practice, while serial tasks with high organization benefitted more from whole practice (Fontana et al., 2009). It seems that even within the same evaluation, results on whether whole or part practice is more beneficial differed.

Overall, it can be derived that the most optimal method of learning is uncertain. It is therefore important to consider the cognitive underpinnings that surrounding the differences in whole versus part practice. When considering those, the discrete sequence production (DSP) task is suitable, as it is specifically designed to investigate the underlying mental processes occurring when performing motor tasks (Abrahamse et al., 2013).

1.3. Dance-Step Discrete Sequence Production Task

In order to examine whether progressive part practice indeed provides learning performance advantages over whole practice, this thesis focused on utilising the DSP task to investigate the motor sequence learning. In the usual DSP task, keyboard keys are pressed according to a fixed pattern seen on a computer screen, (Abrahamse et al., 2013). The screen shows representations of all keys, which light up in a specific order. As a response to this the participant is required to press the associated keys in the aforementioned order (Abrahamse et al., 2013). Generally, such a task consists of sequences of 3-7 stimuli that are learned in whole practice. Across the process of learning, the different responses are performed together in chunks (or segments) of 3-5 items (Abrahamse et al., 2013). The task consists of a practice phase, in which the building blocks and internal representations of the sequence are developed and stabilised (Abrahamse et al., 2013). Following that, the internal representations of the sequence are evaluated by comparing its response pattern against that of a novel sequence of movement (Abrahamse et al., 2013). It was established that the task shows a concatenation pattern that suggests that a six-item sequence is processed as two smaller chunks when learning the whole sequence at once (Verwey & Abrahamse, 2012).

Recently, a Dance Step version of the DSP task coined as the DS-DSP task was successfully developed for assessing motor learning (Wiechmann, 2021). In contrast to the key pressing DSP task performed with the hands, the movements of the DS-DSP task are performed with the feet, on a commercially available dance pad. Furthermore, the task also provides higher complexity. Instead of having a different effector for each different stimulus, two feet are responsible for four different stimuli, requiring the individual to optimize effector use themselves. This means full body engagement is necessary in order to reach the best performance on this task.

Since the task contains low levels of organization and a fixed level of complexity, the arguments of whole versus part practice (Naylor & Briggs, 1963) can be examined in detail. Additionally, the DS-DSP task is serial in nature, making it eligible for testing Schmidt and Wrisberg's (2008) assumptions, i.e. using part practice over whole practice on serial tasks.

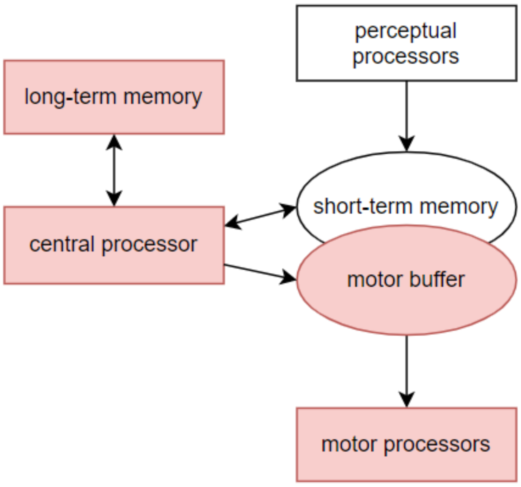
1.4. The Cognitive Framework for Sequential Motor Behaviour

The Cognitive Framework for Sequential Motor Behaviour (C-SMB) was developed by Verwey et al. (2015) on the basis of the findings of the DSP task using whole practice. In contrast to other models explaining motor learning, this model does not concern itself with the specific neural networks involved in motor processing. It rather tries to explain the different stages involved in motor learning based on the cognitive perspective (Verwey et al., 2015). It encompasses multiple other frameworks to make a more coherent paradigm (Keele et al., 2003; Rosenbaum et al., 1986; Rosenbaum et al., 1995; Rosenbaum et al., 2001; Schmidt, 1975; Sternberg et al., 1978; Verwey, 2001; Verwey et al., 2015).

The model assumes that perceptual processors for auditory, visual, and proprioceptive procedures provide input for learning a sequence (Verwey et al., 2015). All of the non-motor input is then stored in short-term memory, in addition to central-symbolic representations which are more abstract in nature (such as task goals) (Verwey et al., 2015). In contrast to that, the motor buffer is responsible for storing specific movement input, which is managed in such a way that additional processing is hardly necessary. This motor buffer contains the so-called motor chunks, which are sequences of movement between 3-5 elements. These two systems are managed by the central processor, which accesses additional information from long-term memory, and then organizes all aspects of the task in order to forward the exact movement into the motor buffer. It is assumed that short-term memory and motor buffer are not entirely separated, but rather facilitated by parallel processing. It is therefore possible for more effortful, slow processing to occur at the same time as more automatic processing (Verwey et al., 2015). The central processor manages this system including all parallel processes, while also having an active role in the learning process. Overall, this leads to the motor chunks already being loaded into the motor buffer while the recognition process is still taking place, making the overall process swifter and more efficient.

Figure 2

Schematic of the C-SMB's Processors



Note. The red background shows the proposed faster and more automatic processing that develops with training and chunking mode. Adapted from "A cognitive framework for explaining serial processing and sequence execution strategies," by W. B. Verwey, C. H. Shea, and D. L. Wright, 2015, *Psychonomic Bulletin and Review, 22*(1), p. 59 (https://doi.org/10.3758/s13423-014-0773-4). Copyright 2015 by the Psychonomic Society, Inc.

During early learning, individuals perform movements in the reaction mode, as it is responsible for processing unfamiliar stimuli (Verwey & Abrahamse, 2012). Within this mode, movement is purely based on the related stimulus, requiring lots of cognitive effort in addition to continuous input from perceptual processors (Verwey & Abrahamse, 2012). Once a sequence of movement is learned, the motoric areas of the brain which are responsible for chunking take over. More direct connections between sequence recognition and motor execution result in the processing happening in an automatic, faster fashion (see Figure 2) (Verwey et al., 2015). In the next processing stage, the motor processors begin executing the movements stored in the motor buffer by cycling through the information until all movements have been executed (Verwey et al., 2015).

When automaticity develops, it can be observed that the first movement's response is slower, as the stimulus is still being identified, which stands in contrast to the faster response time on the following movements (Verwey & Abrahamse, 2012). According to the authors, this discrepancy can be seen as the result of the identification process of the movement sequence that is still in-progress during the first movement. However, after the recognition process is over and the first movement has cued the sequence, the following steps are then able to be processed at a much faster rate and the chunking mode is developed and utilised s (Verwey & Abrahamse, 2012).

It is important to consider that the motor buffer only holds a capacity of 3-5 steps at once. Thereby the automaticity of the chunking mode generally cannot grasp motor learning sequences longer than that, meaning that longer sequences are typically split into multiple chunks (Verwey & Abrahamse, 2012). This causes a slowed down response in the middle of a learning block, when the second chunking block must be loaded into the motor buffer (Verwey & Abrahamse, 2012).

When the chunking mode is inhibited, another process by which learning may occur takes over. The so-called associative mode is inferior to the chunking mode, as it is much slower, with no large improvement in execution time after training (Verwey & Abrahamse, 2012). In the associative mode, a stimulus becomes associated with its predecessor, and this association functions as a way in which to learn (Verwey & Abrahamse, 2012). This means that each part of the sequence of stimuli is loaded into the motor buffer separately, and one movement must be performed before the next one is loaded. Since the chunking mode is much more efficient considering it can load multiple movements into the motor buffer at once, it is used as a first mode of learning, with the association only being activated when the chunking mode is exhausted (Verwey & Abrahamse, 2012).

1.5. Progressive Motor Chunking

Importantly, the observations of learning patterns in the C-SMB framework have typically been made using whole practice. Within this type of practicing, the chunking capabilities are too small to encompass the whole task, leading to interruptions in the learning flow. Based on this, the question emerges whether a progressive introduction of the motor sequence might facilitate learning, as the cognitive processes involved may have better ability to sort the information efficiently.

The reaction mode which occurs in the beginning of learning places high mental demands on the learner since it requires lots of input. However, if the sequence to be learned encompasses only very few items in the beginning, it is much easier to shift from reaction mode to chunking mode. This is the case as a short sequence can easily be processed in one single motor chunk, leading to efficient and fast processing outright (Verwey & Eikelboom, 2003). Furthermore, this also means that following the first block of practice, individuals must only engage in reaction mode when encountering the single progressively added step. Since previous blocks of learning have already organized the preceding movements into chunks (or into associations), the cognitive processor would only require very little time to adapt to the newly introduced movement. Overall, this should result in more efficient cognitive processing of the sequence overall.

Additionally, the progressive chunking means that the first part of the movement sequence is practiced extensively since the beginning. The involvement of fewer items may also have benefits on memory. It should be much easier for that part of the sequence to be transferred into long-term memory than it would if the sequence was practiced as a whole from the beginning. This in turn aids the cognitive processor. Since it is more efficient to pull information from long-term memory, the cognitive processor has less executive workload within the first part of the sequence. This would mean that automaticity can develop easily, freeing cognitive capacities for the later introduced parts of the sequence. This study therefore aims to utilize this mechanism in the process of learning.

In addition, this automatization in the first few steps should lead to better chunking ability, as the cognitive processor is able to optimize them more quickly. Chunks can also be developed according to the unique movement pattern which develops across the progression, meaning that the chunks may be more flexible in their development. Potentially, chunks can easily adopt more steps along the way, finding the most efficient chunking pattern independent of sequence length.

Lastly, even if the chunking mode cannot be utilised and individuals have to revert back to association mode, a benefit of progressive chunking would remain; the intuition here is that associations between the different movements are more easily developed due to a smaller sequence load, especially because of the first few movements. The following movements as the sequence progresses can easily be adapted into that schema. Associations between responses are therefore stronger, aiding in faster and smoother learning times.

1.6. The Current Study

The current experiment aims to examine whether progressive part practice facilitates learning in the DS-DSP task. To that end, participants follow a six-block training session in which progressively more items are introduced into the sequence. After that, they are confronted with the familiar learned sequence, or a new unfamiliar one, in order test their performance further. This is done on the basis of the C-SMB framework's concept of motor chunking, and established assumptions about whole versus part practice. Overall, it is assumed that progressive learners obtain clear advantages over whole learners in their cognitive processing of the task. Therefore, the following hypothesis was examined: progressive learners perform significantly better in comparison to whole learners when executing a DS-DSP task. Better performance can hereby be measured on various dimensions. For one, the accuracy of the sequences may be considered, both across the process of learning as well as during testing against a novel sequence. Furthermore, response time on the accurate sequences is also of relevance across the learning phase and during testing against an unfamiliar sequence. Additionally, it is also important to consider the response time pattern at the individual step level; that way it is possible to determine underlying patterns of concatenation within the learning phase.

2. Methods

2.1. Participants

Twenty-four participants were recruited via the University of Twente's mandatory study participation credit system, as well as over social media. They were not informed about the aims of the experiment until after the study was completed. In order to ensure randomization, the participants were assigned to the different conditions in an alternating pattern according to the order in which they participated. The participants had to meet certain exclusion criteria, namely they were not allowed to have consumed any alcohol in the last 24 hours and they were not supposed to suffer from any physical and mental impairments that might affect performance.

Table 1

Frequencies of Demographic Variables

Participant Characteristic	Progressive Learners		Whole Learners	
	M	*SD*	*M*	*SD*
Age	21.2	2.3	21.8	2.7
Height (in cm)	173.0	10.9	178.0	9.7
Weight (in kg)	71.0	12.3	72.9	13.3

Note. Progressive Learners n = 12. Whole Learners n = 12.

Table 2

Descriptive Statistics of Demographic Variables

Participant characteristic	Progressive Learners		Whole Learners	
	n	*%*	*n*	*%*
Gender				
Female	8	66.7	3	25.0
Male	4	33.7	9	75.0
Smoking participants	0	0.0	2	16.7
Preferred Foot				
Left	0	0.0	2	16.7
Right	11	91.7	9	75.0
Both	1	8.3	1	8.3

Note. Progressive Learners n = 12. Whole Learners n = 12.

Of all participants, 13 were male and 11 were female, and the mean age was 21.5 (SD = 2.4. Participants were on average 175cm (SD = 10.4) tall and weighed 71.9kg (SD = 12.5).

Overall, the vast majority preferred their right foot (83.3%). The left foot or no preferred foot each was indicated by 8.3% of the participants. Additionally, 29.2% of participants had prior experience with DSP tasks, and 79.2% indicated some motor sequence learning experience based on their private activities (e.g. playing the piano). Participants indicated they played console games in their free time (58.3%), and the average time spent was 9.2 hours (SD = 8.5) per week. These participants mostly indicated intermediate gaming skills.

2.2. Measures

2.2.1. Demographic Questions

The beginning block of questions concerned demographic information of the participants, such as age, gender, which of their feet is dominant, their height and their weight. Furthermore, they were asked about whether they smoke, and whether they have consumed alcohol in the last 24 hours.

2.2.2. Affect Grid

The Affect Grid is a single item measure investigating a person's current emotions along the two dimensions pleasure and arousal (Russell et al., 1989). Overall, there is strong discriminant validity between the two dimensions (Russell et al., 1989; Scott Killgore, 1998), and significant convergent validity with other measures of mood (Scott Killgore, 1998). It was used in order to track participants' mood and potentially detect adverse effects of the task on the participants.

2.2.3. NASA Task Load Index

The NASA Task Load Index (TLX) is a scale designed to measure the mental workload of a task. It measures said concept on the basis of six dimensions: mental demand, physical demand, temporal demand, performance, effort, and frustration. Overall, the NASA TLX has proven to have excellent reliability, with a Cronbach's alpha coefficient of over 0.80 (Xiao et al., 2005). Furthermore, great convergent validity was also established, based on a positive Person's correlation between the NASA TLX and the SWAT and WP scales ($p<.001$) (Rubio et al., 2004). In this study, it has provided an opportunity to specifically track the demands that were placed on the participants.

2.2.4. Post practice survey

The questionnaire presented after the practice phase of the experiment concerned the recall abilities of the participant. They were asked to note the sequences they have seen previously, as well as the strategy they used to recall the sequences. Additionally, they were asked further questions regarding their experience with sequence experiments, as well as experience with sequence learning (such as gaming, playing an instrument, or dancing) in general.

2.3. The Dance-Step Discrete Sequence Production Task

The device used to conduct the study was the 'D-Force Non-Slip Dance Pad', which can be connected to a computer via USB. It was 36.5 × 32 inches in size. The different arrows on the pad were set up to correspond to different keys on a keyboard (pointing left corresponding to 'a', pointing right corresponding to 'd', pointing up corresponding to 'w', and pointing down corresponding to 's'). The stimuli were presented in E-Prime® 2.0 Software (Psychology Software Tools Inc., Sharpsburg, USA) on a 24-inch LG FLATRON W224422PE DFC full HD monitor with a screen refresh setting of 60hz.

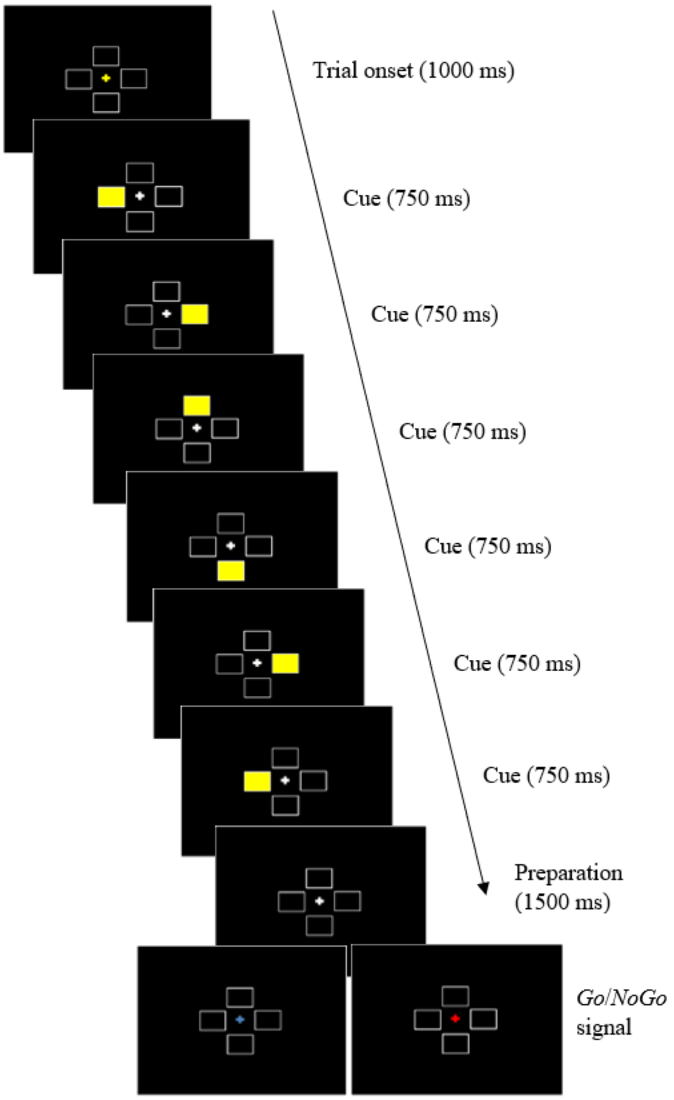
The sequences that were performed on the dance pad were counterbalanced by rotating the steps within each sequence. Overall, this resulted in a full counterbalancing rotation over eight participants. The task on the dance mat consisted of six practice/learning blocks and two testing blocks, one of which showed the familiar sequence again, and another one which showed a novel sequence not before seen by the participant. Each block consisted of 48 sequences, and each sequence entailed six steps. Per block, there were always two sequences shown in random order. In the experimental chunking condition, the sequence was not shown at once, but was introduced progressively. This means that for the first block, the participant only practiced the first three steps, for the second block the first four steps were practiced, blocks three and four entailed the first five steps, and finally in blocks five and six all six steps were practiced together. In contrast, participants in the control condition saw all six steps for all six practice rounds. After that, the participants completed the two testing blocks, one of which was equal to the practiced sequence, and one of which was novel to the participant.

2.4. Procedure

After signing the informed consent form, the participants filled out their first questionnaire while the researcher left the room. They were asked general demographic questions, as well as given versions of the Affect Grid and the NASA TLX items. They were instructed to answer questions asking about a task by referring to them sitting in a chair and filling out the questionnaire. The last question they were asked referred to their weight. The participants were instructed to take of their shoes and remove all other heavy items from their body. Then the participants weighed themselves on the provided scale and the accurate weight was recorded.

Figure 2

An Example of the Sequence shown, including the Go/NoGo Signal



Note. The arrow descriptions indicate the amount of time each stimulus was shown. Adapted from “The Discrete Sequence Production Task in the Form of a Step Task: An Application of Individual Exponential Learning Curves in Motor Sequence Learning” by E. Wiechmann, 2021, University of Twente, p. 11 (retrieved from https://essay.utwente.nl/87430/1/Wiechmann_BA_BMS.pdf).

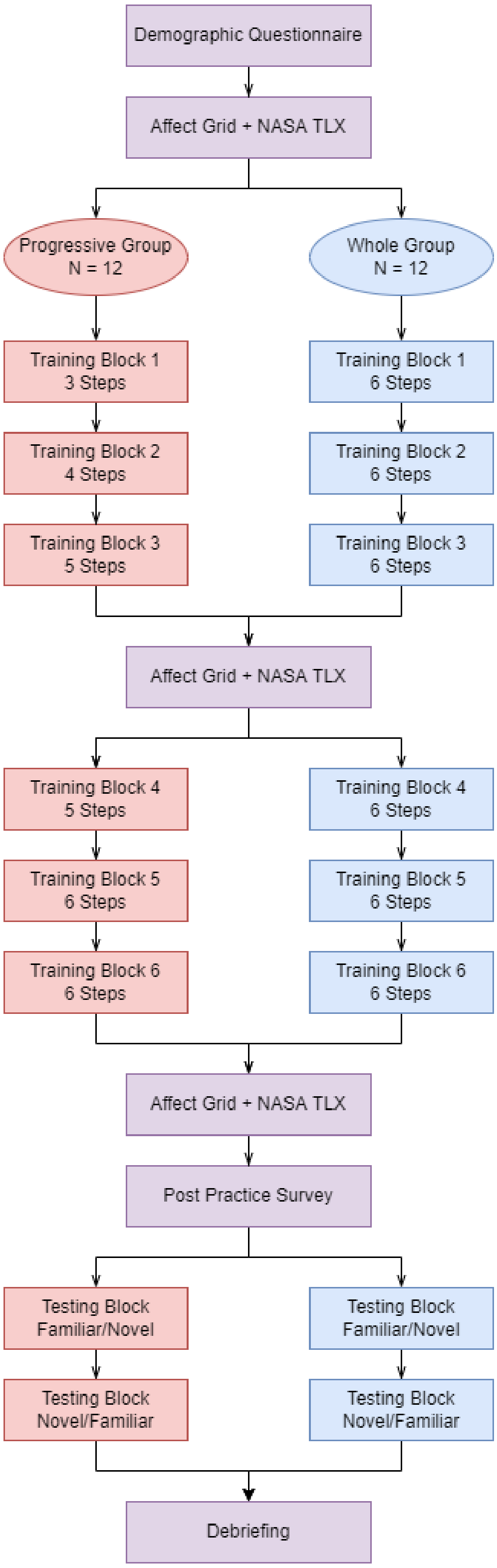
Following that, the participants stepped onto the dance pad with their shoes off, where it was controlled that the pad responded appropriately to the participant's steps, and that the monitor was placed approximately at eye level. After that, they were instructed on the way the screen shows the sequences they were about to perform. They were told that there are four rectangles on the screen representing the arrows on the dance pad, which would light up in yellow to show the sequence. After that, the cross in the middle would light up either in blue, meaning they were to repeat the sequence as accurately and fast as possible, or in red, meaning to simply wait for the next sequence to be shown.

When there were no further questions, the participants began with the first block of the DS-DSP task and performed the other seven blocks subsequently. During the blocks, the researcher stayed in the room in order to ensure no technical difficulties would occur. Each block had a 30 second break in the middle, during which the participant was free to stretch and move a little bit. After each block except for block 3 and 6, the participant had a 3-minute break before the next block was started. After block 3, the participant had a 10-minute break during which they completed the Affect Grid and NASA TLX, this time referring to the DSP task, while the researcher left the room once again. Once the participants had completed all six training blocks, they filled out the Affect Grid and NASA TLX questionnaire a last time (again referring to the DSP task), in addition to answering further questions about the sequences they have performed and the way they remembered them while the researcher waited outside the room. This process took approximately five minutes. Next, the participants performed two testing blocks, again with a 3-minute break between them. One of the blocks showed the sequences the participants have already been practicing before, the other block consisted of novel sequences.

After completing all sequences, the participants were debriefed regarding the purpose of the experiment and were given a small snack as a thank you.

Figure 3

Schematic of the Procedure



Note. The purple background indicates procedures that were performed equally by all participants. The red background shows the procedures performed by the progressive learners. The blue background provides the procedure of the whole learners. Between most blocks, a 3-minute break was taken before the next one was introduced. The breaks between blocks three and four and six and seven were ten and five minutes long, respectively. The last two blocks were counterbalanced with participants receiving either the familiar or the novel block first.

2.5. Data Analysis

The performance data was analysed via linear mixed effect regression (LMER) models using the lme4 package Version 1.1-29 (Bates et al., 2015) in RStudio Version 2022.02.2. The study was conducted as a between groups factorial design, and multiple factors and variables were analysed. An advantage of the LMER model over a traditional Analysis of Variance (ANOVA) lies in the fact that it allows for accounting subjects as random factor. This means that the within-group effect can be considered on an individual basis. Overall, this leads to more accurate results, since clustering in the data is accounted for and it is ensured that group effects can be properly considered (van den Berg, 2021). Furthermore, type of data remains flexible; time of block may be taken as categorical or numerical, depending on better fit of the model (van den Berg, 2021). In this case, considering time as categorical resulted in a better fit of the model.

2.5.1. Accuracy of Blocks

At first, block accuracy served as an outcome variable of performance. It was measured as the percentage of sequences that were fully correct within each block. Accuracy was examined in two separate analyses. In the first phase, the learning accuracy was predicted based on Group (i.e. chunking versus control group) and Block (i.e. blocks one to six). Since the progressive learning group only practiced all six steps in the last two learning blocks, an additional analysis was run based on Group (i.e. chunking versus control group) and Block (i.e. block 5 versus block 6). In the second phase, the testing phase, accuracy was predicted by Group (i.e. chunking versus control group) and Sequence Familiarity (i.e. familiar versus unfamiliar sequence).

2.5.2. Mean Response Time of Blocks

Mean response time per block of accurate sequences was also examined as the dependent variable. Prior to analysis, the data was adjusted in such a way that only the fully accurate sequences were considered. Furthermore, any sequence that had a total execution time 2.5 times the overall mean of the block was excluded to sort out outliers. Overall, this resulted in 14.9 % of sequences being excluded in the learning phase, and 20.3% of sequences being excluded in the testing phase.

The response time during the learning phase was predicted by Group (i.e. chunking versus control group) and Block (i.e. blocks one to six). Again, since only blocks five and six are equal in length across the two groups, an additional analysis was run with the independent variables Group (i.e. chunking versus control group) and Block (i.e. block 5 versus block 6). The testing phase was analysed with the predictors Group (i.e. chunking versus control group) and Sequence Familiarity (i.e. familiar versus unfamiliar sequence).

2.5.3. Mean Response Time of Steps

Lastly, an additional analysis was run in order to examine the concatenation pattern of the response time at a step level. Since only blocks five and six are of equal length stepwise, only that part of the learning phase was examined. Mean response time per step per block per participant was examined as the outcome variable. The independent variables were Group (i.e. chunking versus control group) Block (i.e. block 5 versus block 6) and Step (i.e. steps one to six). Additional analyses were performed to tease out the locus of interaction. The outcome variable step level response time was predicted by Group (i.e. chunking versus control group) and Step (i.e. steps one to three), for both block 5 and block 6 separately. Furthermore, two analyses with the independent variables Group (chunking versus control group) and Step (i.e. steps four to six) were conducted, for block 5 and 6, respectively.

3. Results

3.1. Accuracy of Blocks

In order to evaluate the performance of the two groups, accuracy during both learning and testing phases served as the first indication of performance.

Table 2

Effects of Accuracy per Block Models

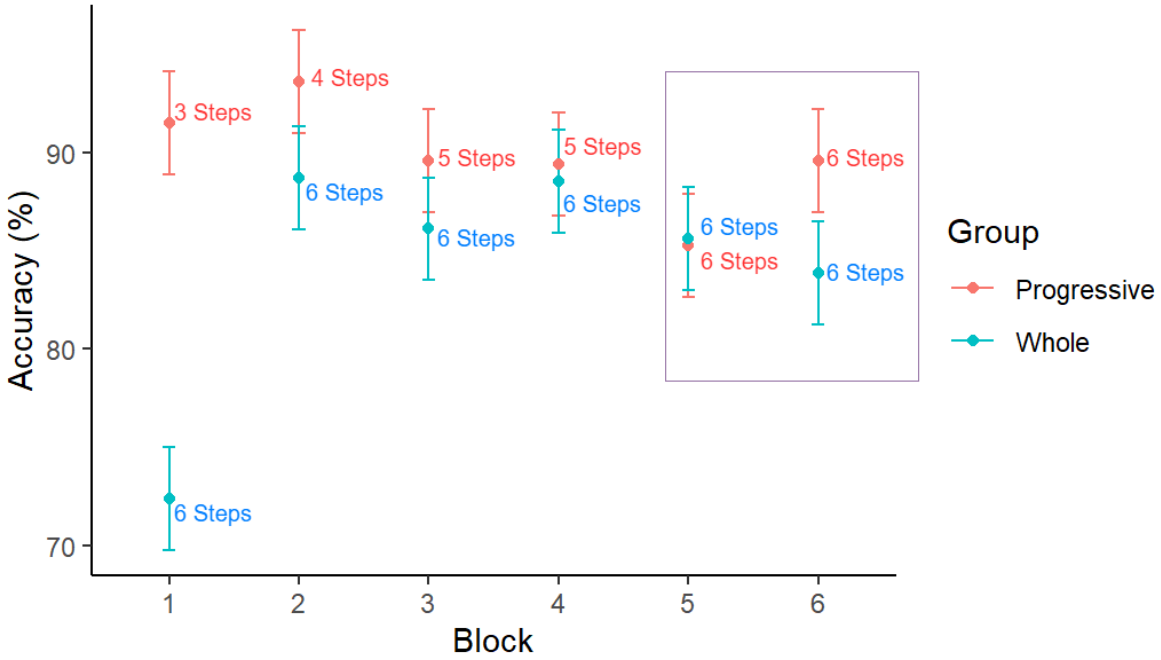
Analysis	Effect	df	χ^2	Significance
Accuracy Training (all blocks)				
	Group	1	3.75	
	Block	5	31.60	***
	Group x Block	5	38.74	***
Accuracy Training (block 5 & 6 only)				
	Group	1	0.44	
	Block	1	0.73	
	Group x Block	1	4.00	*
Accuracy Testing Phase				
	Group	1	0.04	
	Familiarity	1	6.89	**
	Group x Familiarity	1	0.64	

Note. N = 24 (n = 12 for each condition).

* $p < .05$, ** $p < .01$, *** $p < .001$

Figure 4

Accuracy as predicted by Block and Group during the Learning Phase



Note. Single data points are labelled to indicate the number of steps taken in the block. Blocks five and six are highlighted to emphasize the comparison between the equal number of steps.

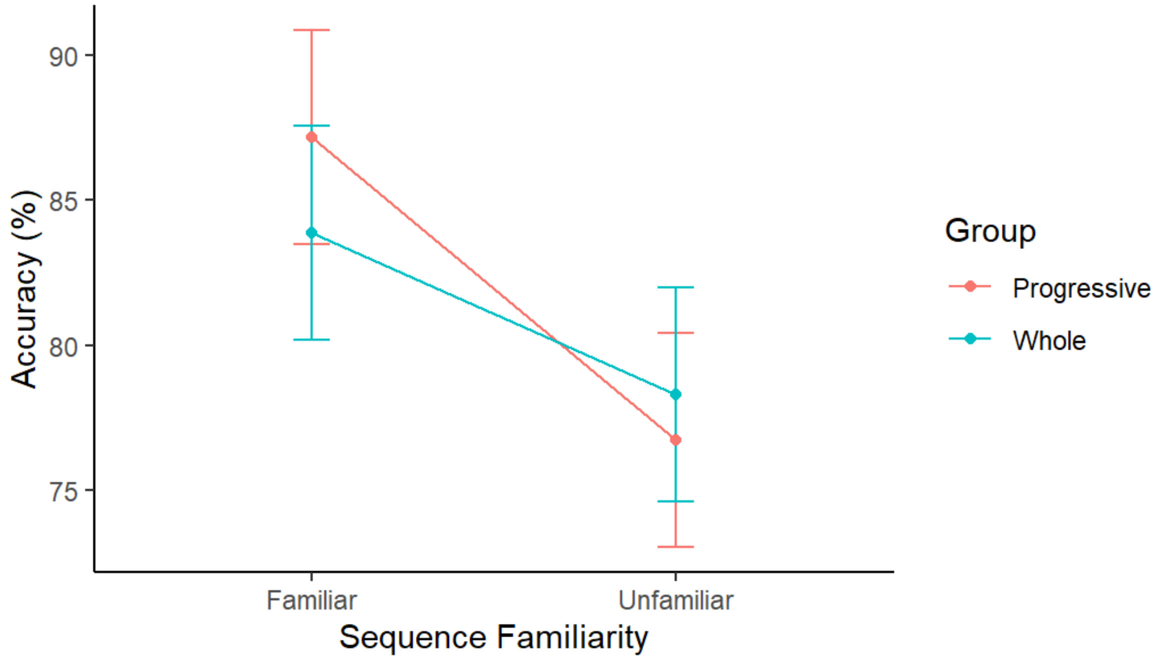
The training accuracy model revealed an only marginally significant main effect of Group (p = .053) but a significant main effect of Block, showing gradual increase of accuracy across session of learning. Additionally, the Group x Block interaction also proved to be significant, reflecting that across blocks, the predicted slope of the two groups is significantly different. This, in addition to the plot (Figure 4) indicates that progressive learners performed significantly more accurate than whole learners.

In the last two training blocks, where both groups received 6-item sequences, the model revealed no significant main effects of Group (p = .507) and Block (p = .392) but did show significance in the Group x Block interaction in the predicted slopes. When considering the plot (see purple rectangle in Figure 4) it becomes obvious that progressive learners have significantly improved their accuracy in comparison to whole learners.

Overall, the results of training accuracy suggest a significantly better performance of progressive learners, confirming the hypothesis.

Figure 5

Comparison of Accuracy on the two Testing Blocks as a Function of Group



Note. The plot shows the non-significant interaction of Group and Sequence Familiarity. The contrast between the familiar and the unfamiliar sequence is significant.

In addition to modelling training accuracy, a model was performed in order to compare accuracy across the learned sequence with accuracy across an unfamiliar sequence. In this model, no significant effects were found for Group (p = .838) or the Familiarity x Group interaction (p = .424). The main effect of Familiarity merely showed the difference in accuracy between repeating a familiar sequence and learning a new one (see Figure 5). According to the testing accuracy, the hypothesis must therefore be rejected.

3.1. Response Time of Blocks

Another measure of performance that was examined is response time of accurate sequences.

Table 3

Effects of Response Time per Block Models

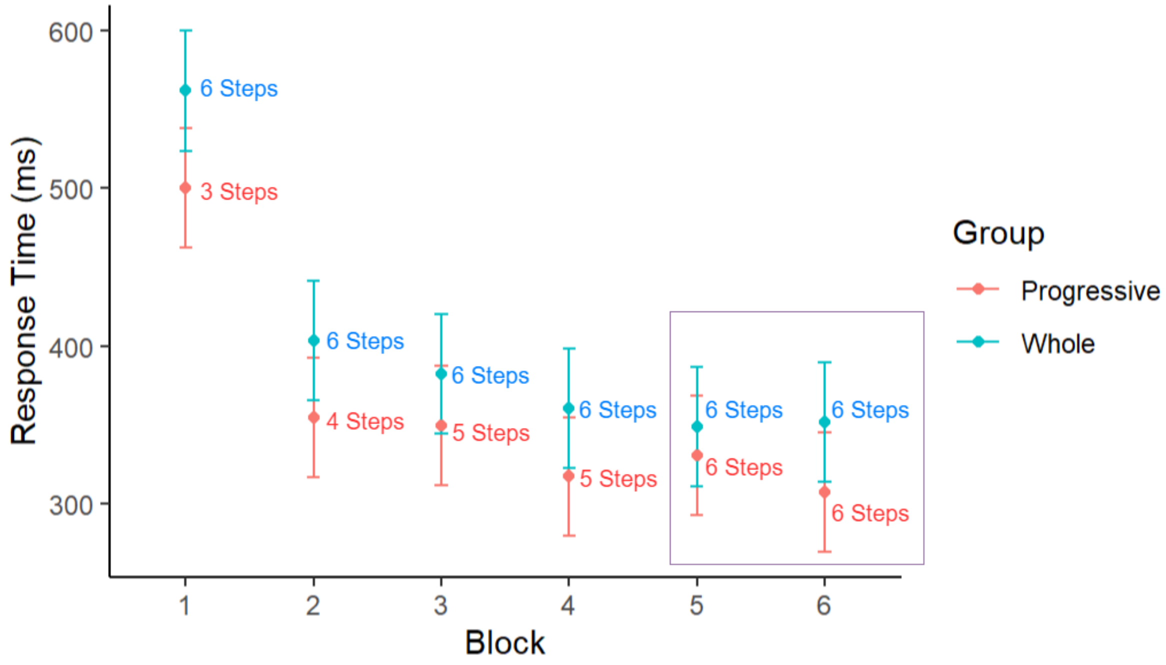
Analysis	Effect	df	χ^2	Significance
Response Time Training (all blocks)				
	Group	1	0.44	
	Block	5	2448.63	***
	Group x Block	5	23.96	***
Response Time (blocks 5 & 6 only)				
	Group	1	0.22	
	Block	1	12.86	***
	Group x Block	1	15.14	***
Response Time Testing				
	Group	1	0.14	
	Familiarity	1	194.67	***
	Group x Familiarity	1	1.13	

Note. N = 24 (n = 12 for each condition).

* $p < .05$, ** $p < .01$, *** $p < .001$

Figure 6

Mean Response Time as predicted by Block and Group during the Learning Phase



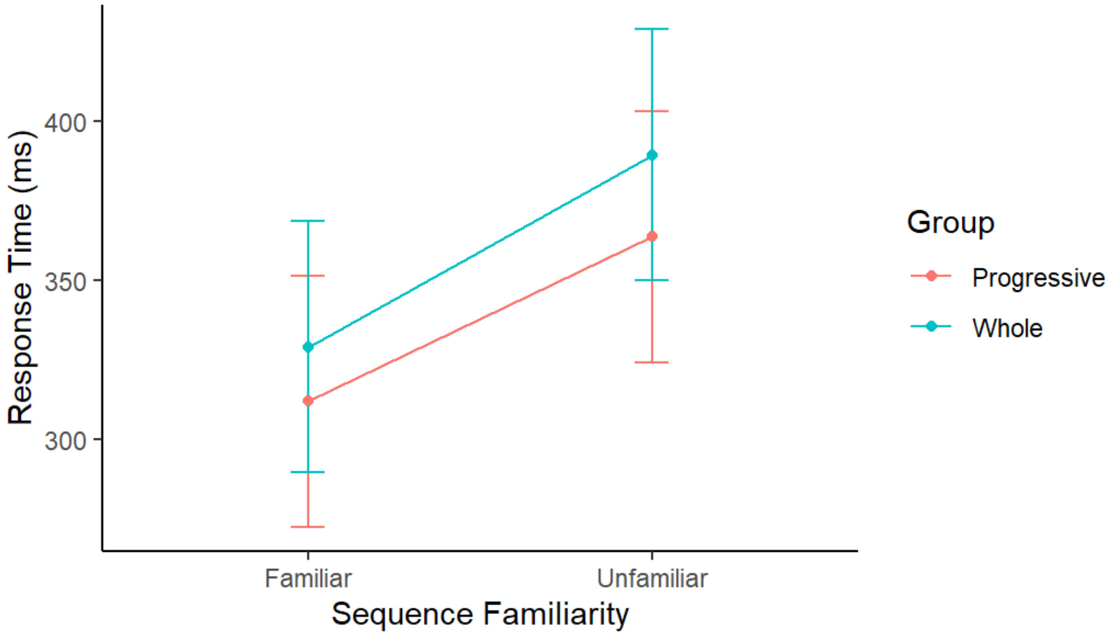
Note. Single data points are labelled as to indicate the number of steps taken in the block. Blocks five and six are highlighted to emphasize the comparison between the equal number of steps.

The response time model of learning revealed no main effects for Group (p = .436), but a main effect for Block was found, which clearly indicates an improvement of response time over the course of learning (see Figure 6). Additionally, a significant interaction of Group x Block shows that overall, progressive learners learned faster in comparison to whole learners.

In the last two training blocks where both groups received 6-item sequences, the model revealed no significant main effect for Group (p = .638). However, there were significant effects for both Block, and the interaction Group x Block. The plot shows that progressive learners did indeed learn the full sequence faster than whole learners (see purple rectangle in Figure 6). Overall, the results of the training block response time therefore confirm the hypothesis that progressive learners perform better than whole learners.

Figure 7

Comparison of Response Time on the two Testing Blocks as a Function of Group



Note. The plot shows the relatively equal slopes of Group and Sequence Familiarity. The contrast between the familiar and the unfamiliar sequence is significant.

The model of the familiar sequence in contrast to an unfamiliar one (see Figure 7) showed that overall, there were no effects for Group (p = .703) or the interaction Group x Familiarity (p = .288). There was a main effect of Familiarity, which only shows that there is indeed a difference in response time when performing a previously learned sequence versus an entirely novel one. Based on the testing, the hypothesis of better performance of progressive learners must therefore be rejected.

3.3. Response Time of Steps

When considering the previous results, it is still not clear whether progressive and whole learners perform equally, or whether progressive learning leads to better performance. Therefore, in order to further understand the cognitive underpinnings behind the performance, the performance between the two Groups was examined at the step level with the outcome variable of step response time.

Table 4

Effects of Response Time per Step Models only considering Blocks Five and Six

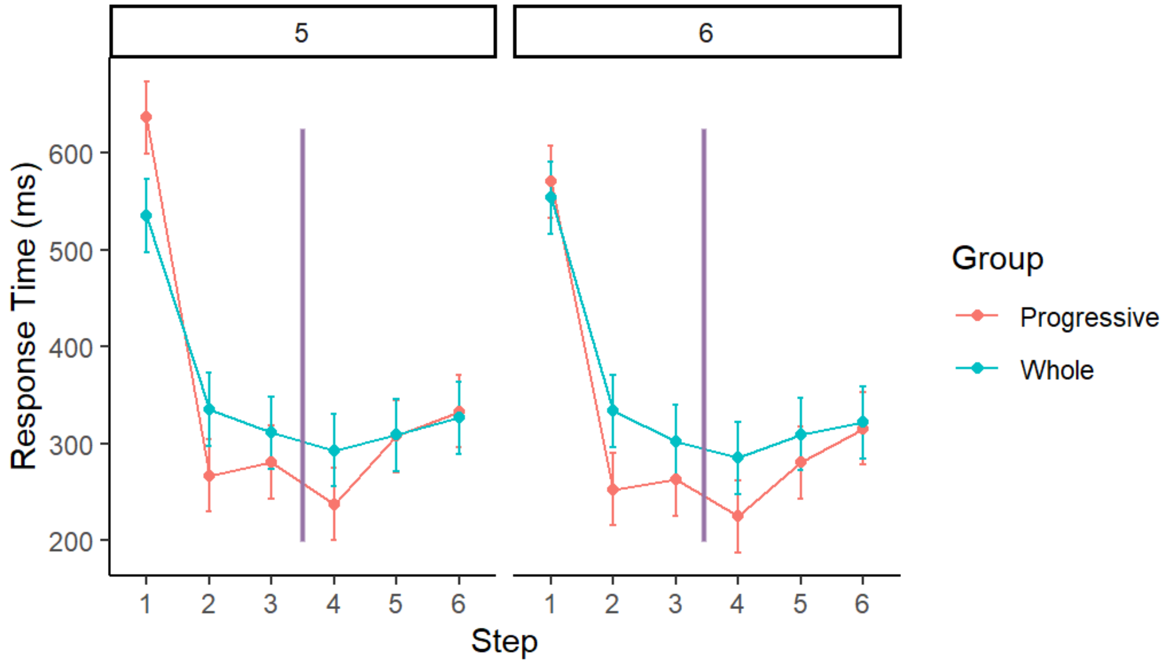
Analysis	Effect	df	χ^2	Significance
Response Time Step Level				
	Group	1	0.16	
	Group x Block	1	10.62	**
	Group x Step	5	124.40	***
	Group x Block x Step	5	12.56	*
Response Time Steps 1-3 Block 5				
	Group	1	0.00	
	Step	2	1187.04	***
	Group x Step	2	85.65	***
Response Time Steps 1-3 Block 6				
	Group	1	0.41	
	Step	2	901.95	***
	Group x Step	2	21.33	***
Response Time Steps 4-6 Block 5				
	Group	1	0.08	
	Step	2	47.12	***
	Group x Step	2	12.47	**
Response Time Steps 4-6 Block 6				
	Group	1	0.63	
	Step	2	92.10	***
	Group x Step	2	15.97	***

Note. N = 24 (n = 12 for each condition).

* $p < .05$, ** $p < .01$, *** $p < .001$

Figure 8

Response Time at a Step Level as a Function of Group per Block

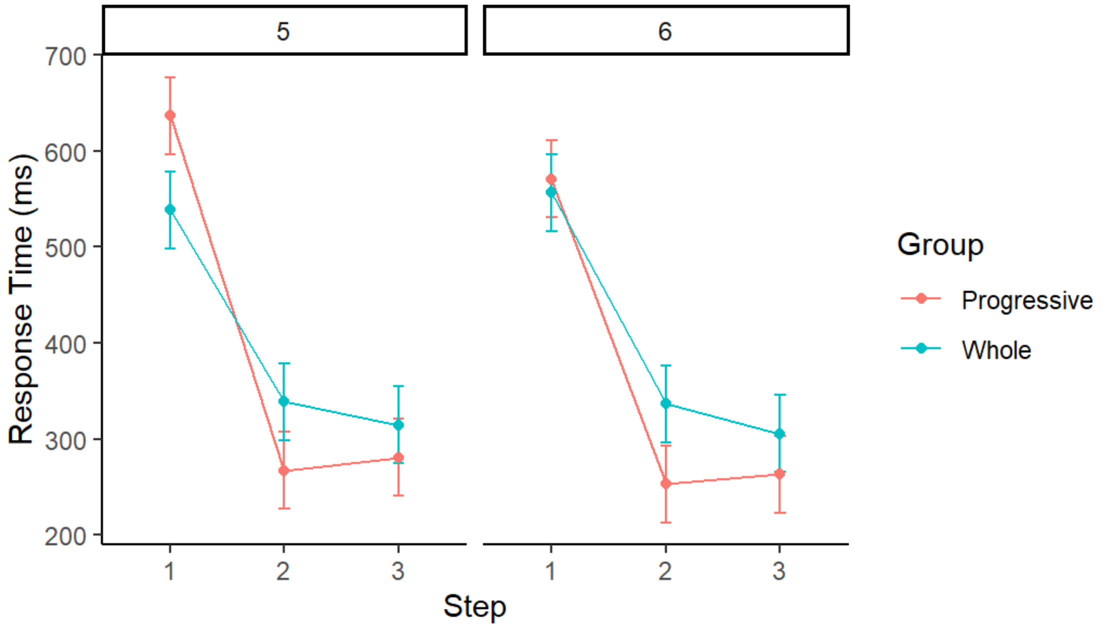


Note. Visualization of the significant Group interactions with Step and Block. The purple line illustrates the contrast in the pattern between the two halves of the sequence. For a more detailed view of the two halves, see Figure 9 and 10.

The model on response time revealed no significant main effect for Group (p = .688). There was a Group x Block interaction that shows a significant difference of the two groups across the two comparable learning blocks, and the Group x Step interaction reveals significant differences between the groups at a step level. In combination with the plot (see Figure 8) it is clearly evident that the progressive learners performed significantly faster across the sequence. Additionally, it also becomes evident that the progressive group still experienced a significant improvement from Block 5 to Block 6, especially across steps one and five. Furthermore, the interaction Group x Block x Step also reveals that the difference in response time has changed significantly depending on all three variables.

Figure 9

Response Time in the First Half of the Sequence across Groups

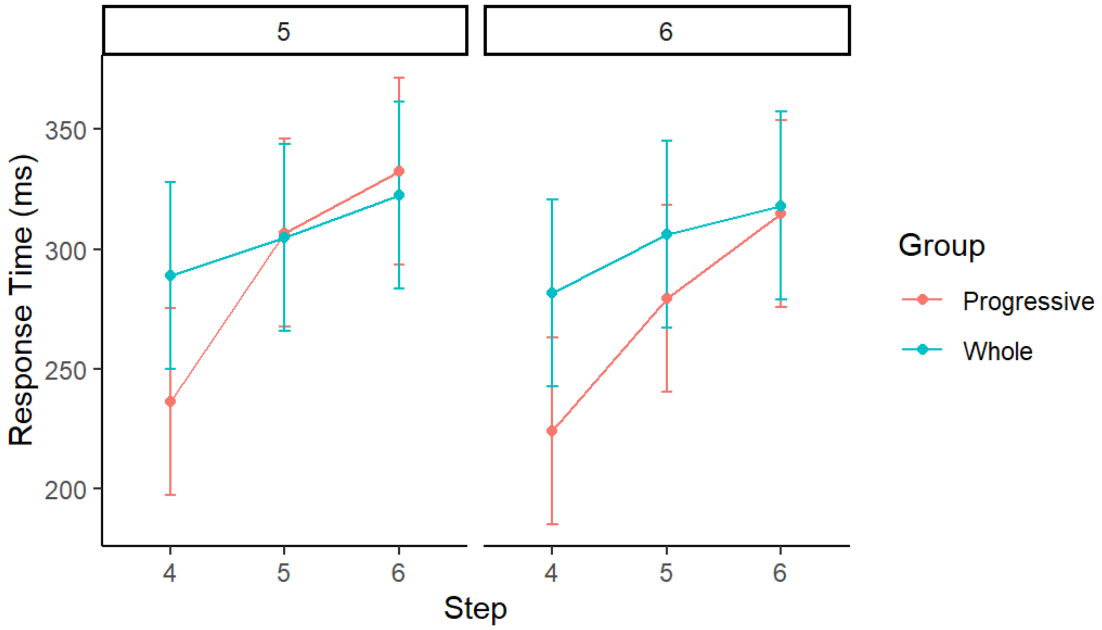


Note. Visualization of the significant interactions between the first steps of the sequence and Group. The main point of interaction lies at step two.

In order to further tease out the locus of interactions, four additional analyses were run to examine whether the effect remains also when looking at the data in a more detailed manner. When modelling only steps one to three for each block separately, no Group effect has been found in either analysis (p = .963 for Block 5, p = .522 for Block 6). Both Block 5 and Block 6 had a main effect for Step. Additionally, the interaction Group x Step is also significant in both Block 5, and in Block 6. The plot (Figure 9) furthermore illustrates that the interaction mainly stems from the rapid decrease in response time of step two for the progressive learners. This shows that the difference between the group interaction remains significant when only considering parts of the overall model.

Figure 10

Response Time in the Second Half of the Sequence across Groups



Note. Visualization of the significant interactions between the last steps of the sequence and Group. The main point of interaction lies at step four.

The same pattern was observed when considering only steps four to six in the two separate blocks. There was no main effect for Group in either block (p = .780 and p = .630), but the main effect for Step was significant in both blocks. Additionally, the Group x Step interaction was also significant in Block 5, and in Block 6, meaning that the interaction remained when looking at the last steps isolated. When considering the plots (Figure 10), it appears that the main cause of the interaction effect stems from step four. Steps five and six (especially in Block 5) are much more similar across groups.

4. Discussion

Overall, the results support the prediction of a beneficial effect due to progressive learning on multiple levels. During the training phase, performance showed that progressive learning is not only a more accurate way of learning, but also leads to significantly faster execution of the sequence. This also means that no trade-off was made between speed and accuracy within progressive learning, since consistently better performance occurred on both measurements.

Generally speaking, our results fit the consensus of Naylor and Briggs (1968) in that we categorised the DS-DSP task as one that requires low organization and is high in complexity, making it ideal for progressive chunking. Furthermore, the task is serial in nature, meaning it is also in line with Schmidt and Wrisberg's (2008) assumptions about whole versus part practice. Overall, it appears that progressive learning induces a different cognitive organization compared to whole practice during a low-organization serial task that leads to better and faster execution of the learned motor sequence. This fits nicely with the assumption of a less difficult reaction phase when first learning a new sequence, resulting in a more efficient mode of processing. Figure 4 clearly indicates high levels of accuracy from the beginning for the progressive learners thanks to lower step counts, which remains higher than the accuracy of the whole learners even when adding new steps to the sequence. Similarly, Figure 6 shows the decrease in response time which also always remains lower for the progressive learners, even when they are always receiving new input.

However, this effect does not seem to have significant influence when contrasting it against an unfamiliar sequence of steps. When considering solely the comparison between the learned sequence and a new one, there is no difference between the two learning groups, neither in accuracy nor in response time. This means that when the sequence has been properly learned, the representations and loading of the sequence into the motor buffer seems to happen in equal efficiency, leading to equal performance. These conclusions are in line with the conflicting results of previous studies. It has been found before that learning in part versus learning as a whole does not make a significant difference on the final performance of the task, even if mean effect sizes may differ across groups (Fontana et al., 2009; Hansen & Tremblay, 2005). Overall, it seems that the process of learning itself is organized in a different way depending on the group, thus leading to a difference in cognitive organization of the steps, rather than in the overall performance of each trial.

This consideration of differences in cognitive organization is plausible when examining the response time on a step level. It has been found that the response time pattern between the whole versus progressive learners was indeed significantly different. Figure 8 shows a relatively stable response time for whole learners after the first step has been initiated. This suggests that whole learners did not actually develop a typical concatenation pattern like previous literature would suggest (Abrahamse et al., 2013; Verwey et al., 2015). Rather, the pattern indicates that similar response times occur between all steps, pointing towards the same cognitive process happening over again; associative processing has been favoured (Verwey & Abrahamse, 2012). After the first step determines the sequence, the associated next steps are retrieved in a similar cognitive process until the full sequence has been executed. Since the cognitive process of retrieving each step separately from long-term memory and loading it into the motor buffer is both slower and more effortful, it is clear why the whole learning group performed significantly more slowly across steps than the progressive learning group. Generally, it seems like the processing is more like that of a Serial Reaction Time (SRT) task, which is also typically performed in association mode (Abrahamse et al., 2010; Jiménez, 2008; Jiménez et al., 2011). This is further confirmed when looking at a comparable footstep version of said task, which also found no chunking tendency during learning (Du & Clark, 2017).

An entirely different pattern presents itself for the progressive learners. After the first step has been processed, the following ones are performed significantly faster than the ones in the whole learning group (see Figure 10). A small concatenation point was evident at step three, which is promptly followed by a much faster executed step four. Generally, the concatenation pattern is typical for the traditional DSP task (Abrahamse et al., 2013; Verwey & Eikelboom, 2003). This points towards the development of motor chunks, meaning multiple movements are loaded into the motor buffer at once, hence leading to faster response times in progressive over whole learners (Verwey et al., 2015). Based on this, it can be derived that the main advantage of progressive learning lies in the concatenation of the first four steps and the speed gained by their efficiency and automaticity.

It is obvious that the DS-DSP task entails much more complexity than the traditional DSP task. During the classic DSP task, individuals are required to merely press different keys with different fingers. Each finger represents a different stimulus and must be pressed accordingly. In contrast, the DS-DSP task requires individuals to decide for themselves which foot to use for which stimulus, in addition to what is fastest considering the next step in the sequence. Those considerations overall require much more motor information to be encompassed within one single step (i.e. direction of moving the body, direction of moving the foot, pressing the correct arrow). This additional complexity thereby means that whole learners were entirely unable to chunk the steps, while chunking for progressive learners occurred at a smaller scale. However, there is one inconsistency in this pattern; the last two steps of the sequence have a much longer reaction time than the others, similar to the whole learning group. This contrasts with a typical concatenation pattern (typically, reaction time should stay at approximately the same level as the first step after the concatenation point). Therefore, different reasonings must be considered.

An alternative explanation for the significant difference in response time between the groups would be that the progressive group used a combination of chunking mode and association mode in order to execute the movements. It makes sense to assume that both the chunking mode as well as the association mode explain different parts of the response time pattern. The first part of the sequence is processed in two chunks. After that, steps five and six are processed in associative mode. This explanation would consider the stark increase of response time towards the last few steps; since they have not been practiced nearly as long as the first ones, automaticity could not be developed to the same degree, which means they require more effort by the cognitive processor and longer loading times. This is inconsistent with the explanation expressed in the introduction, as it was assumed that chunking may entail a certain degree of flexibility to adjust for additional steps. However, it seems like the opposite is the case: once a solid chunking pattern has developed within the first few steps, chunking stops, and the novel steps are processed in association mode. This may be due to the smaller amount of practice for the last steps. Regardless, in contrast to the traditional DSP task, it appears that the DS-DSP task does not entail chunking as a default pattern, again highlighting a similarity to results in SRT tasks (Jiménez, 2008; Jiménez et al., 2011); also those using footsteps (Du & Clark, 2017). Additionally, in SRT, chunking may occur under special circumstances (Jiménez, 2008; Jiménez et al., 2011). Possibly, a similar assumption can be made about the DS-DSP task; progressive learning may be a special circumstance which allows for chunking in said task.

In conclusion, the results of this study indicate a stark contrast in cognitive processing between the whole and the progressive learners, which results in beneficial learning effects during the training phase. However, during testing, the different types of cognitive organization appear to perform equally well. Based on these conclusions, the hypothesis of better performance in progressive learners on a low-organization serial task may be partially accepted.

4.1. Real Life Implications and Future Directions

Overall, it is clear that progressive chunking leads to a better learning performance for new learners by making the process of learning itself less cognitively demanding in low-organization tasks. This can mean that learners process the new sequence more easily, leading to an overall more enjoyable learning experience. This is important, as fun acts as a moderator to motivation and performance on a new task (Chan et al., 2019). Furthermore, the learned sequence itself may be easier to perform, thanks to the chunking and automaticity during the first few steps. These implications may be relevant for any low-organization serial learning task using feet, such as aerobics, dancing, martial arts, or specific running and jumping patterns utilized in other types of sports.

In the future, it is advisable to expand the sequence, in order to make it more complex. This would show whether the positive learning effects of progressive introduction of steps transfer to increased complexity. It would allow to further explore the organization of chunks across a longer sequence, and whether different concatenation patterns emerge. Additionally, further complexity might also eventually lead to other ways that the progressive learning group outperforms whole learners.

5. Acknowledgements

I would like to thank Rian Telman and Nerijus Bertalis for the excellent collaboration in data collection. A special thank you also goes to Tim Segger for his assistance in data handling. Lastly, I would like to thank my supervisor Dr. Russell W. Chan for his great assistance and support throughout this process.

References

Abrahamse, E. L., Jiménez, L., Verwey, W. B., & Clegg, B. A. (2010). Representing serial action and perception. *Psychonomic Bulletin and Review*, *17*(5), 603–623. https://doi.org/10.3758/PBR.17.5.603

Abrahamse, E. L., Ruitenberg, M. F. L., de Kleine, E., & Verwey, W. B. (2013). Control of automated behavior: Insights from the discrete sequence production task. *Frontiers in Human Neuroscience*, *7*, 1–16. https://doi.org/10.3389/fnhum.2013.00082

Adams, J. A. (1987). Historical Review and Appraisal of Research on the Learning, Retention, and Transfer of Human Motor Skills. *Psychological Bulletin*, *101*(1), 41–74. https://doi.org/10.1037/0033-2909.101.1.41

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using {lme4}. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Chan, S. C. H., Wan, J. C. L., & Ko, S. (2019). Interactivity, active collaborative learning, and learning performance: The moderating role of perceived fun by using personal response systems. *International Journal of Management Education*, *17*(1), 94–102. https://doi.org/10.1016/j.ijme.2018.12.004

Doyon, J. (2008). Motor sequence learning and movement disorders. *Current Opinion in Neurology*, *21*(4), 478–483. https://doi.org/10.1097/wco.0b013e328304b6a3

Doyon, J., Gabitov, E., Vahdat, S., Lungu, O., & Boutin, A. (2018). Current issues related to motor sequence learning in humans. *Current Opinion in Behavioral Sciences*, *20*, 89–97. https://doi.org/10.1016/j.cobeha.2017.11.012

Du, Y., & Clark, J. E. (2017). New insights into statistical learning and chunk learning in implicit sequence acquisition. *Psychonomic Bulletin and Review*, *24*(4), 1225–1233. https://doi.org/10.3758/s13423-016-1193-4

Fontana, F. E., Furtado, O., Mazzardo, O., & Gallagher, J. D. (2009). Whole and part practice: A meta-analysis. *Perceptual and Motor Skills*, *109*(2), 517–530. https://doi.org/10.2466/PMS.109.2.517-530

Hansen, S., & Tremblay, L. (2005). Part and whole practice: Chunking and online control in the acquisition of a serial motor task. *Research Quarterly for Exercise and Sport*, *76*(1), 60–66. https://doi.org/10.1080/02701367.2005.10599262

Jiménez, L. (2008). Taking patterns for chunks: Is there any evidence of chunk learning in continuous serial reaction-time tasks? *Psychological Research*, *72*(4), 387–396. https://doi.org/10.1007/s00426-007-0121-7

Jiménez, L., Méndez, A., Pasquali, A., Abrahamse, E., & Verwey, W. (2011). Chunking by colors: Assessing discrete learning in a continuous serial reaction-time task. *Acta Psychologica*, *137*(3), 318–329. https://doi.org/10.1016/j.actpsy.2011.03.013

Keele, S. W., Mayr, U., Ivry, R., Hazeltine, E., & Heuer, H. (2003). The Cognitive and Neural Architecture of Sequence Representation. *Psychological Review*, *110*(2), 316–339. https://doi.org/10.1037/0033-295X.110.2.316

Lingyun, L. (2021). Reform thinking brought by progressive public physical education teaching Under the concept of " educate people in the whole process ". *5th International Conference on Economics, Management Engineering and Education Technology*, 380–385. https://doi.org/10.25236/icemeet.2021.079

Naylor, J. C., & Briggs, G. E. (1963). Effects of task complexity and task organization on the relative efficiency of part and whole training methods. *Journal of Experimental Psychology*, *65*(3), 217–224. https://doi.org/10.1037/h0041060

Rosenbaum, D. A., Loukopoulos, L. D., Vaughan, J., Meulenbroek, R. G. J., & Engelbrecht, S. E. (1995). Planning reaches by evaluating stored postures. *Psychological Review*, *102*(1), 28–67. https://doi.org/10.1037/0033-295X.102.1.28

Rosenbaum, D. A., Vaughan, J., Meulenbroek, R. J., & Jansen, C. (2001). Posture-based motion planning: Applications to grasping. *Psychological Review*, *108*(4), 709–734. https://doi.org/10.1037/0033-295X.108.4.709

Rosenbaum, D. A., Weber, R. J., Hazelett, W. M., & Hindorff, V. (1986). The parameter remapping effect in human performance: Evidence from tongue twisters and finger fumblers. *Journal of Memory and Language*, *25*(6), 710–725. https://doi.org/10.1016/0749-596X(86)90045-8

Rubio, S., Díaz, E., Martín, J., & Puente, J. M. (2004). Evaluation of Subjective Mental Workload: A Comparison of SWAT, NASA-TLX, and Workload Profile Methods. *Applied Psychology*, *53*(1), 61–86. https://doi.org/10.1111/j.1464-0597.2004.00161.x

Russell, J. A., Weiss, A., & Mendelsohn, G. A. (1989). Affect Grid: A Single-Item Scale of Pleasure and Arousal. *Journal of Personality and Social Psychology*, *57*(3), 493–502. https://doi.org/10.1037/0022-3514.57.3.493

Schmidt, R. A., & Wrisberg, C. A. (2008). *Motor learning and performance: A situation-based learning approach*. Human Kinetics.

Schmidt, Richard A. (1975). A schema theory of discrete motor skill learning. *Psychological Review*, *82*(4), 225–260. https://doi.org/10.1037/h0076770

Scott Killgore, W. D. (1998). The affect grid: A moderately valid, nonspecific measure of pleasure and arousal. *Psychological Reports*, *83*(2), 639–642. https://doi.org/10.2466/pr0.1998.83.2.639

Sternberg, S., Monsell, S., Knoll, R. L., & Wright, C. E. (1978). The Latency and Duration of Rapid Movement Sequences: Comparisons of Speech and Typewriting. In *Information Processing in Motor Control and Learning* (Issue December). https://doi.org/10.1016/b978-0-12-665960-3.50011-6

van den Berg, S. (2021). *Analysing data using linear models*. University of Twente.

Verwey, W. B. (2001). Concatenating familiar movement sequences: The versatile cognitive processor. *Acta Psychologica*, *106*(1–2), 69–95. https://doi.org/10.1016/S0001-6918(00)00027-5

Verwey, W. B., & Abrahamse, E. L. (2012). Distinct modes of executing movement sequences: Reacting, associating, and chunking. *Acta Psychologica*, *140*(3), 274–282. https://doi.org/10.1016/j.actpsy.2012.05.007

Verwey, W. B., & Eikelboom, T. (2003). Evidence for lasting sequence segmentation in the discrete sequence-production task. *Journal of Motor Behavior*, *35*(2), 171–181. https://doi.org/10.1080/00222890309602131

Verwey, W. B., Shea, C. H., & Wright, D. L. (2015). A cognitive framework for explaining serial processing and sequence execution strategies. *Psychonomic Bulletin and Review*, *22*(1), 54–77. https://doi.org/10.3758/s13423-014-0773-4

Walker, M. P., Brakefield, T., Seidman, J., Morgan, A., Hobson, J. A., & Stickgold, R. (2003). Sleep and the time course of motor skill learning. *Learning and Memory*, *10*(4), 275–284. https://doi.org/10.1101/lm.58503

Wiechmann, E. (2021). *The Discrete Sequence Production Task in the form of a Step Task: An application of Individual Exponential Learning Curves in Motor Sequence Learning*. University of Twente.

Wu, W. F. W., Young, D. E., Schandler, S. L., Meir, G., Judy, R. L. M., Perez, J., & Cohen, M. J. (2011). Contextual interference and augmented feedback: Is there an additive effect for motor learning? *Human Movement Science*, *30*(6), 1092–1101. https://doi.org/10.1016/j.humov.2011.02.004

Wulf, G., & Schmidt, R. A. (1997). Variability of practice and implicit motor learning. *Journal of Experimental Psychology: Learning Memory and Cognition*, *23*(4), 987–1006. https://doi.org/10.1037/0278-7393.23.4.987

Xiao, Y., Wang, Z., Wang, M., & Lan, Y. (2005). The appraisal of reliability and validity of subjective workload assessment technique and NASA-task load index. *中华劳动卫生职业病杂志 [Chinese Journal of Industrial Hygiene and Occupational Diseases]*, *23*(3), 178—181. http://europepmc.org/abstract/MED/16124892

Appendix A

Cognitive States and Motor Learning Questionnaire

Start of Block: Demographic questions

*

Age How old are you?

____

Handedness Which of your feet is dominant one? (With which leg do you kick the ball?)

- Right-foot (1)
- Left-foot (2)
- Comfortable with both feet (3)

Q24 What is your gender?

- Male (1)
- Female (2)
- Other (3)
- Prefer not to say (4)

Smoking Do you smoke?

- Yes (1)
- No (2)

Alcohol Did you drink alcohol in the last 24 hours?

- ◯ Yes (1)
- ◯ No (2)

---

*

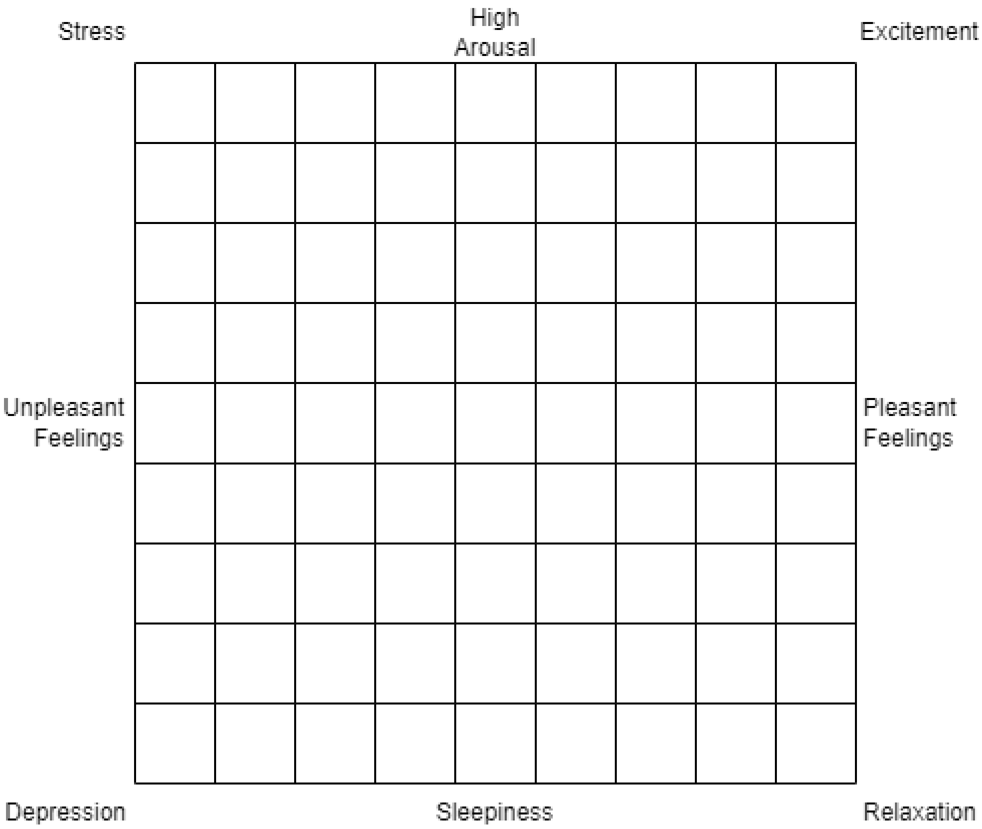
Height How tall are you (in cm)?

____

End of Block: Demographic questions

Start of Block: Affect T1

Please mark on the grid with the mouse how you currently feel.



End of Block: Affect T1

Start of Block: NASA TLX 1

Mental demand. How mentally demanding was the task?

very low very high

1 21

move slider to indicate ()	

Physical demand. How physically demanding was the task?

very low very high

1 21

move slider to indicate ()	

Temporal demand. How hurried or rushed was the task?

very low very high

1 21

move slider to indicate ()	

Performance. How successful were you in accomplishing what you were asked to do?

very low very high

1 21

move slider to indicate ()	

Effort. How hard did you have to work to accomplish your level of performace?

	very low	very high
	1	21
move slider to indicate ()		

Frustration. How insecure, discouraged, irritated, stressed and annoyed were you?

	very low	very high
	1	21
move slider to indicate ()		

End of Block: NASA TLX 1

Start of Block: Weight

What is your weight in kg?

____

End of Block: Weight

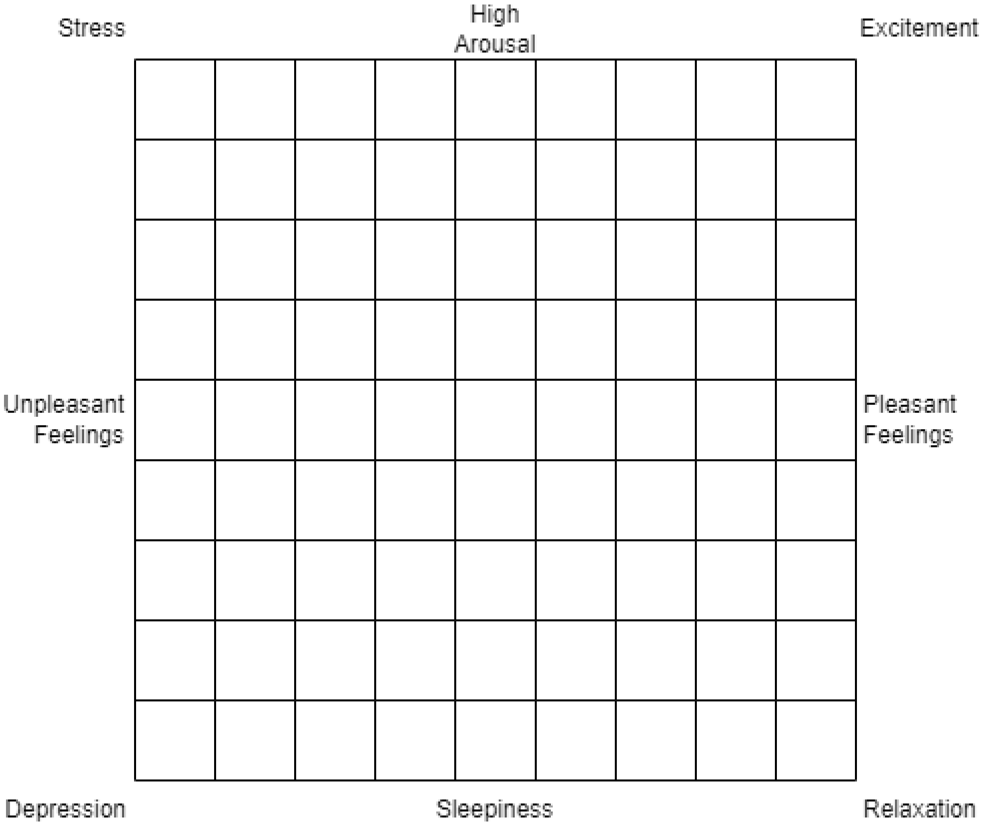
Start of Block: T2

Please stop here and wait for further instructions. Please leave this page open until the experimenter tells you to continue.

End of Block: T2

Start of Block: Affect T2

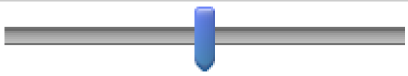
Please mark on the grid with the mouse how you currently feel.



End of Block: Affect T2

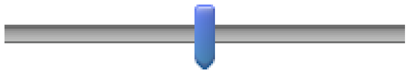
Start of Block: NASA TLX 2

Mental demand. How mentally demanding was the task?

	very low	very high
	1	21
move slider to indicate ()		

Physical demand. How physically demanding was the task?

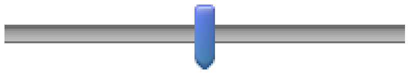
	very low	very high
	1	21

move slider to indicate ()	

Temporal demand. How hurried or rushed was the task?

very low very high

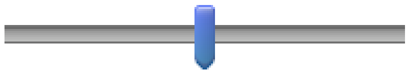
1 21

move slider to indicate ()	

Performance. How successful were you in accomplishing what you were asked to do?

very low very high

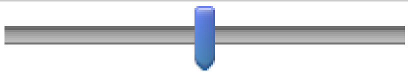
1 21

move slider to indicate ()	

Effort. How hard did you have to work to accomplish your level of performace?

very low very high

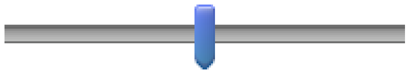
1 21

move slider to indicate ()	

Frustration. How insecure, discouraged, irritated, stressed and annoyed were you?

very low very high

1 21

move slider to indicate ()	

End of Block: NASA TLX 2

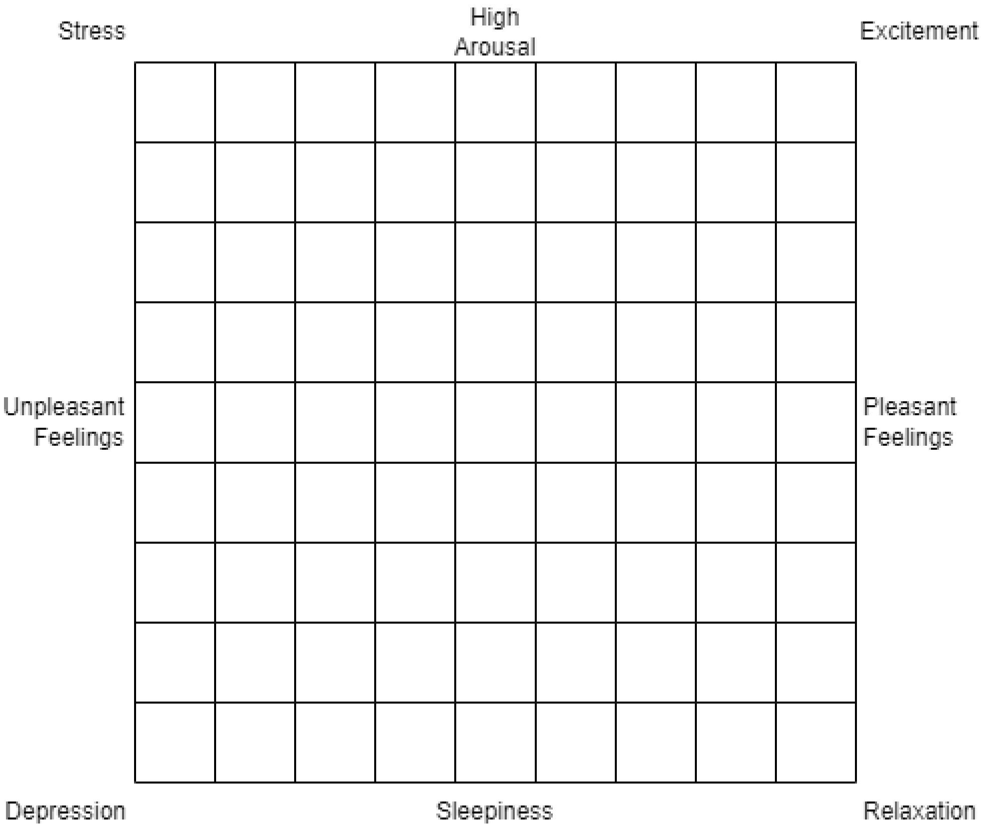
Start of Block: T3

Please stop here and wait for further instructions. Please leave this page open until the experimenter tells you to continue.

End of Block: T3

Start of Block: Affect T3

Please mark on the grid with the mouse how you currently feel.



End of Block: Affect T3

Start of Block: NASA TLX 3

Mental demand. How mentally demanding was the task?

very low very high

1 21

move slider to indicate ()	

Physical demand. How physically demanding was the task?

very low very high

1 21

move slider to indicate ()	

Temporal demand. How hurried or rushed was the task?

very low very high

1 21

move slider to indicate ()	

Performance. How successful were you in accomplishing what you were asked to do?

very low very high

1 21

move slider to indicate ()	

Effort. How hard did you have to work to accomplish your level of performace?

	very low	very high
	1	21
move slider to indicate ()		

Frustration. How insecure, discouraged, irritated, stressed and annoyed were you?

	very low	very high
	1	21
move slider to indicate ()		

End of Block: NASA TLX 3

Start of Block: Survey

What is your participant number? (You can ask the researcher)

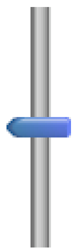
____

In this experiment you reacted by stepping your foot after percieving a stimulus light. There were two main sequences used throughout the experiment. For all two sequences, can you indicate which keys you pressed in correct order? You do not have to recall which sequences came first. You can always ask the researcher for extra explanaition with filling in the following questions.

The first sequence was

	Up (1)	Right (2)	Down (3)	Left (4)
Position 1 (1)				
Position 2 (2)				
Position 3 (3)				
Position 4 (4)				
Position 5 (5)				
Position 6 (6)				

How sure are you about the correctness of the first sequence on a scale of 1 (unsure) to 10 (sure)?

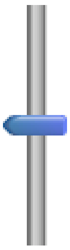


0 (0)
1 (1)
2 (2)
3 (3)
4 (4)
5 (5)
6 (6)
7 (7)
8 (8)
9 (9)
10 (10)

The second sequence was

	Up (1)	Right (2)	Down (3)	Left (4)
Position 1 (1)	○	○	○	○
Position 2 (2)	○	○	○	○
Position 3 (3)	○	○	○	○
Position 4 (4)	○	○	○	○
Position 5 (5)	○	○	○	○
Position 6 (6)	○	○	○	○

How sure are you about the correctness of the second sequence on a scale of 1 (unsure) to 10 (sure)?



0 (0)
1 (1)
2 (2)
3 (3)
4 (4)
5 (5)
6 (6)
7 (7)
8 (8)
9 (9)
10 (10)

End of Block: Survey

Start of Block: Survey 2

How were you able to recognize the sequences?

- ☐ I remembered the order of the arrows (1)
- ☐ I remembered the position of the arrows (2)
- ☐ I remembered the position of the blocks on the screen (3)
- ☐ I tapped the sequence in my mind (4)
- ☐ I re-enacted the sequence with my body (5)
- ☐ In another way, namely: (7) ________________

Have you participated before in an experiment having to do with learning sequences?

- ○ Definitely not (1)
- ○ Probably not (2)
- ○ Might or might not (3)
- ○ Probably yes (4)
- ○ Definitely yes (5)

Do you have any personal experience with learning sequences? (think of playing an instrument)

- ○ Definitely not (1)
- ○ Probably not (2)
- ○ Might or might not (3)
- ○ Probably yes (4)
- ○ Definitely yes (5)

How many hours a week do you spend with console gaming (think of ps4, nintendo switch)?

	0	10	20	30	40	50	60	70	80	90	100
Hours a week spent gaming ()											

Which level gamer would you consider yourself to be?

- Complete beginner/I do not game (1)
- Beginner (2)
- Intermediate (3)
- Advanced (4)
- Expert (5)

Do you have any remarks about this experiment?

____

End of Block: Survey 2

Appendix B

Analyses Progressive Motor Learning

Lia Veith

2022-06-27

Data Setup

```
# set working directory
setwd("C:/Users/liave/Documents/Module 12/R")

# import data
df = read_excel("C:/Users/liave/Documents/Module 12/R/ChunkControl_Datafra
me.xlsx")

# separate training and testing blocks
df_train = df %>% subset(block <7)
df_test = df %>% subset(block >6)
```

Accuracy Model for Training Dataset

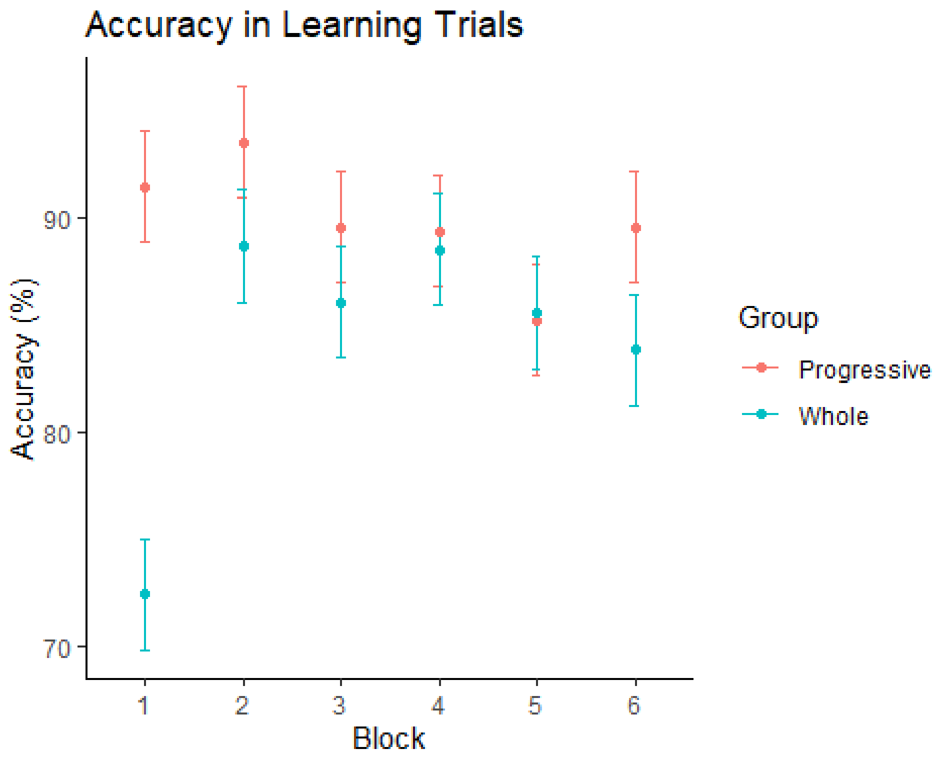
```
# load accuracy data set
dfACC = read_excel("C:/Users/liave/Documents/Module 12/R/ACCURACY.xlsx")

# separate training blocks
df_trainACC = dfACC %>% subset(block <7)

# factors
df_trainACC$subject = factor(df_trainACC$subject)
df_trainACC$group = factor(df_trainACC$group)
df_trainACC$block = factor(df_trainACC$block, ordered = TRUE, levels=c('1
', '2', '3', '4', '5', '6'))

# model training
m.df_train.acc = lmer(percent.ACC ~ group * block + (1|subject), data = df
_trainACC, REML = FALSE)

Anova(m.df_train.acc)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: percent.ACC
##               Chisq Df Pr(>Chisq)
## group        3.7457  1    0.05294 .
## block       31.5967  5  7.140e-06 ***
## group:block 38.7460  5  2.671e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m.df_train.acc, ddf = "Satterthwaite")
```

```
## Linear mixed model fit by maximum likelihood . t-tests use Satterthwait
e's
##   method [lmerModLmerTest]
## Formula: percent.ACC ~ group * block + (1 | subject)
##    Data: df_trainACC
##
##      AIC      BIC   logLik deviance df.resid
##   1010.1   1051.7   -491.1    982.1      130
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3931 -0.5354  0.0325  0.5699  2.3209
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 44.15    6.645
##  Residual             37.94    6.160
## Number of obs: 144, groups:  subject, 24
##
## Fixed effects:
##                        Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)             89.8148     2.0509  24.0000  43.793  < 2e-16 ***
## groupcontrol            -5.6134     2.9004  24.0000  -1.935   0.0648 .
## block.L                 -4.1501     1.7782 120.0000  -2.334   0.0213 *
## block.Q                  1.1555     1.7782 120.0000   0.650   0.5171
## block.C                  3.6880     1.7782 120.0000   2.074   0.0402 *
## block^4                  0.4921     1.7782 120.0000   0.277   0.7824
## block^5                  2.3951     1.7782 120.0000   1.347   0.1805
## groupcontrol:block.L  10.1677     2.5147 120.0000   4.043 9.35e-05 ***
## groupcontrol:block.Q -11.1571     2.5147 120.0000  -4.437 2.04e-05 ***
## groupcontrol:block.C   1.4881     2.5147 120.0000   0.592   0.5551
## groupcontrol:block^4  -3.7731     2.5147 120.0000  -1.500   0.1361
## groupcontrol:block^5   0.8421     2.5147 120.0000   0.335   0.7383
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) grpcnt blck.L blck.Q blck.C blck^4 blck^5 grp:.L gr
p:.Q
## groupcontrl -0.707

## block.L      0.000  0.000

## block.Q      0.000  0.000  0.000

## block.C      0.000  0.000  0.000  0.000

## block^4      0.000  0.000  0.000  0.000  0.000

## block^5      0.000  0.000  0.000  0.000  0.000  0.000

## grpcntrl:.L  0.000  0.000 -0.707  0.000  0.000  0.000  0.000

## grpcntrl:.Q  0.000  0.000  0.000 -0.707  0.000  0.000  0.000  0.000
```

```
## grpcntrl:.C  0.000  0.000  0.000  0.000 -0.707  0.000  0.000  0.000  0.
000
## grpcntrl:^4  0.000  0.000  0.000  0.000  0.000 -0.707  0.000  0.000  0.
000
## grpcntrl:^5  0.000  0.000  0.000  0.000  0.000  0.000 -0.707  0.000  0.
000
##             grp:.C grp:^4
## groupcontrl
## block.L
## block.Q
## block.C
## block^4
## block^5
## grpcntrl:.L
## grpcntrl:.Q
## grpcntrl:.C
## grpcntrl:^4  0.000
## grpcntrl:^5  0.000  0.000
```

```
# training accuracy plot
ae.m.df_train.acc = allEffects(m.df_train.acc)
ae.m.df.df_train.acc = as.data.frame(ae.m.df_train.acc[[1]])

ae.Trainacc = ggplot(ae.m.df.df_train.acc, aes(x=block,y=fit,color=group))
+
  geom_errorbar(aes(ymin=fit-se, ymax=fit+se), width=.1) +
  geom_line() +
  geom_point()+
  ylab("Accuracy (%)")+
  xlab("Block")+
  labs(color = "Group")+
  scale_color_manual(labels = c("Progressive", "Whole"),
                     values = c("#F8766D", "#00BFC4"))+
  ggtitle("Accuracy in Learning Trials")+
  theme_classic()

plot(ae.Trainacc)
```

```
## geom_path: Each group consists of only one observation. Do you need to
adjust
## the group aesthetic?
```

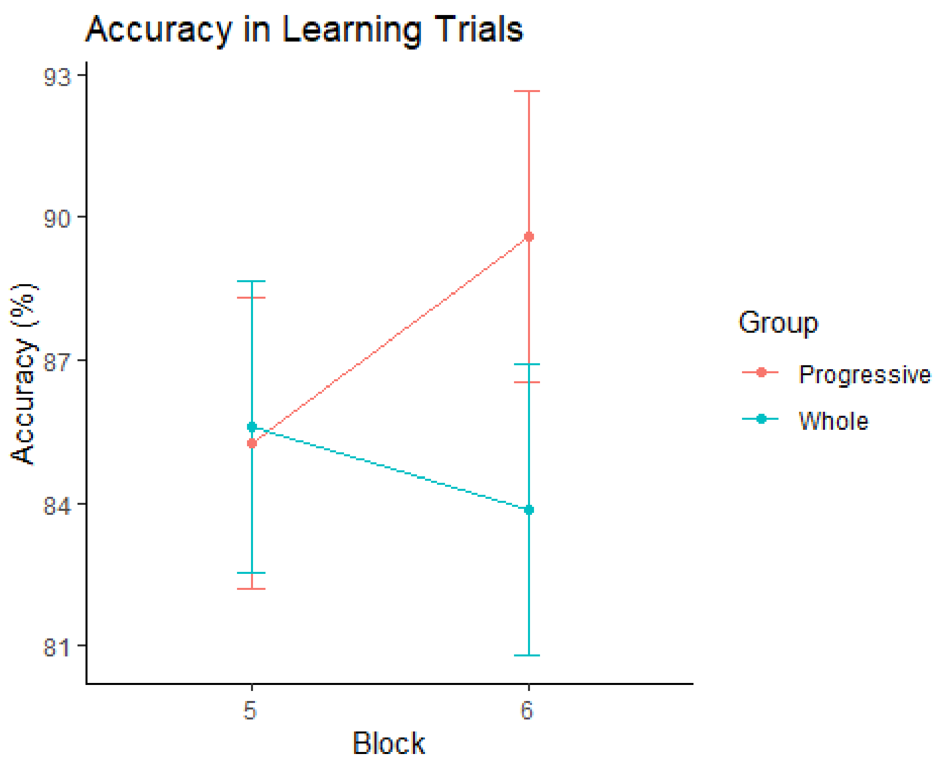



```r
# only retain accurate trials
df_trainacc = subset(df_train, accuracy=="1")
```

Accuracy Model for Training Dataset - only blocks 5 and 6

```r
## additional model training, only comparing blocks 5 and 6

# subset
df_trainACC56 = df_trainACC %>% subset(block >4)

# factors
df_trainACC56$subject = factor(df_trainACC56$subject)
df_trainACC56$group = factor(df_trainACC56$group)
df_trainACC56$block = factor(df_trainACC56$block, ordered = TRUE, levels=c
('1', '2', '3', '4', '5', '6'))

# model training
m.df_train.acc56 = lmer(percent.ACC ~ group * block + (1|subject), data =
df_trainACC56, REML = FALSE)

Anova(m.df_train.acc56)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: percent.ACC
##              Chisq Df Pr(>Chisq)
## group       0.4405  1    0.50688
## block       0.7339  1    0.39162
## group:block 3.9957  1    0.04562 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

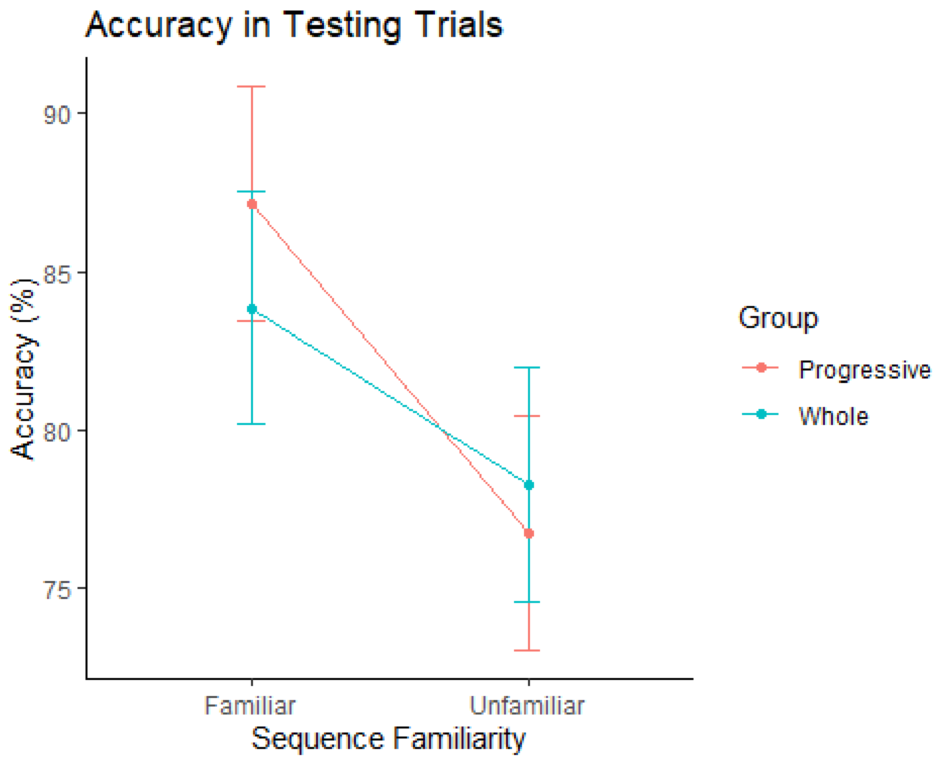
```
summary(m.df_train.acc56, ddf = "Satterthwaite")

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwait
e's
##   method [lmerModLmerTest]
## Formula: percent.ACC ~ group * block + (1 | subject)
##    Data: df_trainACC56
##
##      AIC      BIC   logLik deviance df.resid
##    354.8    366.0   -171.4    342.8       42
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.00777 -0.56780  0.03579  0.53799  1.54761
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 84.77    9.207
##  Residual             27.72    5.265
## Number of obs: 48, groups:  subject, 24
##
## Fixed effects:
##                      Estimate Std. Error     df t value Pr(>|t|)
## (Intercept)            87.413      2.867 24.000  30.490   <2e-16 ***
## groupcontrol           -2.691      4.054 24.000  -0.664   0.5132
## block.L                 3.069      1.520 24.000   2.019   0.0548 .
## groupcontrol:block.L   -4.297      2.149 24.000  -1.999   0.0571 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) grpcnt blck.L
## groupcontrl -0.707
## block.L      0.000  0.000
## grpcntrl:.L  0.000  0.000 -0.707

# training accuracy (blocks 5 and 6) plot
ae.m.df_train.acc56 = allEffects(m.df_train.acc56)
ae.m.df.df_train.acc56 = as.data.frame(ae.m.df_train.acc56[[1]])

ae.Trainacc56 = ggplot(ae.m.df.df_train.acc56,
  aes(x=block, y=fit, color=group, group=group))+
  geom_errorbar(aes(ymin=fit-se, ymax=fit+se), width=.1) +
  geom_line() +
  geom_point()+
  ylab("Accuracy (%)")+
  xlab("Block")+
  labs(color = "Group")+
  scale_color_manual(labels = c("Progressive", "Whole"),
                     values = c("#F8766D", "#00BFC4"))+
  ggtitle("Accuracy in Learning Trials")+
  theme_classic()

plot(ae.Trainacc56)
```



Accuracy Model for Testing Dataset

```
# separate testing blocks
df_testACC = dfACC %>% subset(block >6)

# factors
df_testACC$subject = factor(df_testACC$subject)
df_testACC$group = factor(df_testACC$group)
df_testACC$FAM_UNFAM = factor(df_testACC$FAM_UNFAM)

# model testing
m.df_test.acc = lmer(percent.ACC ~ group * FAM_UNFAM + (1|subject), data =
 df_testACC, REML = FALSE)

Anova(m.df_test.acc)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: percent.ACC
##                   Chisq Df Pr(>Chisq)
## group            0.0420  1   0.837599
## FAM_UNFAM        6.8888  1   0.008674 **
## group:FAM_UNFAM 0.6381  1   0.424403
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m.df_test.acc, ddf = "Satterthwaite")

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwait
e's
##   method [lmerModLmerTest]
## Formula: percent.ACC ~ group * FAM_UNFAM + (1 | subject)
```

```
##    Data: df_testACC
##
##      AIC      BIC   logLik deviance df.resid
##    390.2    401.4   -189.1    378.2       42
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0669 -0.2636  0.2316  0.4798  1.2763
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept)  52.07    7.216
##  Residual             111.10   10.540
## Number of obs: 48, groups:  subject, 24
##
## Fixed effects:
##                               Estimate Std. Error      df t value Pr(>|t
|)
## (Intercept)                     87.153      3.687  43.564  23.635   <2e-1
6 ***
## groupcontrol                    -3.299      5.215  43.564  -0.633   0.530
3
## FAM_UNFAMUNFAM                 -10.417      4.303  24.000  -2.421   0.023
4 *
## groupcontrol:FAM_UNFAMUNFAM    4.861      6.085  24.000   0.799   0.432
2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) grpcnt FAM_UN
## groupcontrl -0.707
## FAM_UNFAMUN -0.583  0.413
## g:FAM_UNFAM  0.413 -0.583 -0.707

# testing accuracy plot
ae.m.df_test.acc = allEffects(m.df_test.acc)
ae.m.df.df_test.acc = as.data.frame(ae.m.df_test.acc[[1]])

ae.Testacc = ggplot(ae.m.df.df_test.acc, aes(x=FAM_UNFAM,y=fit,color=grou
p, group=group))+
  geom_errorbar(aes(ymin=fit-se, ymax=fit+se), width=.1) +
  geom_line() +
  geom_point()+
  ylab("Accuracy (%)")+
  scale_x_discrete("Sequence Familiarity", labels = c("FAM" = "Familiar",
"UNFAM" = "Unfamiliar"))+
  labs(color = "Group")+
  scale_color_manual(labels = c("Progressive", "Whole"),
                     values = c("#F8766D", "#00BFC4"))+
  ggtitle("Accuracy in Testing Trials")+
  theme_classic()

plot(ae.Testacc)
```




```
# only retain accurate trials
df_testacc = subset(df_test, accuracy=="1")
```

Prepare Training Dataset for Response Time Model

```
## calculate the mean value of the block-means

# variables
train_subjects = vector()
train_blocks = vector()
train_means = vector()
train_SD = vector()

# create block means
for (subject in unique(df_trainacc$subject)) {
  for (block in 1:6) {
    current_block_means =
      df_trainacc$meanRT[df_trainacc$subject==subject
                        & df_trainacc$block==block]

    train_subjects = append(train_subjects, subject)
    train_blocks = append(train_blocks, block)
    train_means = append(train_means, mean(current_block_means))
    train_SD = append(train_SD, sd(current_block_means))
  }
}

# create new df with the block means
df_means_train = data.frame(subject=train_subjects,
                            block=train_blocks,
                            block_mean=train_means,
```

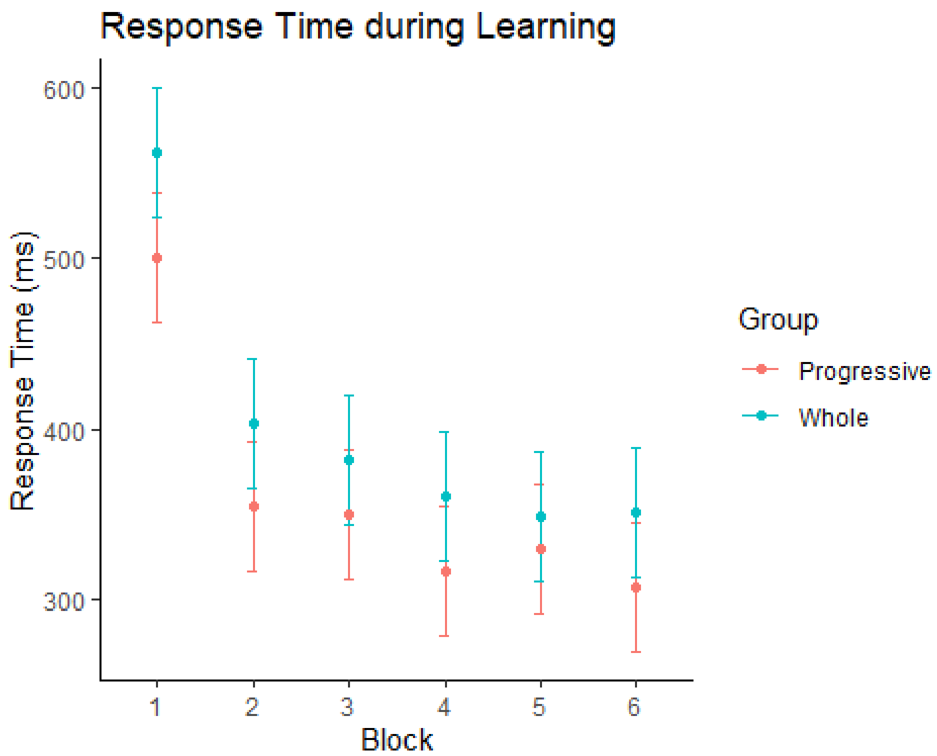
```
                                   block_sd=train_SD)

# filter accurate trials which are no more than 2.5SDs away from the mean
df_trainacc2 = merge(x=df_trainacc, y=df_means_train, by=c("subject", "blo
ck"), sort=FALSE)

df_train1 = df_trainacc2 %>% filter(meanRT < (block_mean+2.5*block_sd))

# retained trials: 5881/6912 = 85.08%

# subset only blocks 5 and 6
df_train1.5.6 = df_train1 %>% subset(block >4)
```

Response Time Model for Training Dataset

```
# factors
df_train1$subject = factor(df_train1$subject)
df_train1$group = factor(df_train1$group)
df_train1$block = factor(df_train1$block, ordered = TRUE, levels=c('1', '2
', '3', '4', '5', '6'))

# model training
m.df_train1.1 = lmer(meanRT ~ block * group + (1|subject), data=df_train1,
 REML = FALSE)
Anova(m.df_train1.1)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: meanRT
##                 Chisq Df Pr(>Chisq)
## block       2448.6252  5  < 2.2e-16 ***
## group          0.6065  1  0.4361054
## block:group   23.9573  5  0.0002213 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m.df_train1.1, ddf="Satterthwaite")

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwait
e's
##   method [lmerModLmerTest]
## Formula: meanRT ~ block * group + (1 | subject)
##    Data: df_train1
##
##      AIC      BIC   logLik deviance df.resid
##  71472.3  71565.8 -35722.2  71444.3     5867
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4419 -0.5228 -0.0875  0.4093  9.1239
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 16928    130.1
##  Residual             10788    103.9
```

```
## Number of obs: 5881, groups:  subject, 24
##
## Fixed effects:
##                        Estimate Std. Error       df t value Pr(>|t|)    
## (Intercept)             359.780     37.606   23.985   9.567 1.17e-09 ***
## block.L                -127.472      4.624 5857.018 -27.568  < 2e-16 ***
## block.Q                  74.622      4.624 5857.003  16.138  < 2e-16 ***
## block.C                 -49.508      4.629 5857.003 -10.695  < 2e-16 ***
## block^4                  16.349      4.632 5856.995   3.530 0.000419 ***
## block^5                 -25.219      4.632 5857.001  -5.444 5.41e-08 ***
## groupcontrol             41.541     53.186   23.990   0.781 0.442410    
## block.L:groupcontrol    -20.273      6.729 5857.053  -3.013 0.002602 ** 
## block.Q:groupcontrol     17.311      6.704 5857.029   2.582 0.009844 ** 
## block.C:groupcontrol      6.194      6.640 5857.025   0.933 0.351004    
## block^4:groupcontrol     10.621      6.597 5857.005   1.610 0.107482    
## block^5:groupcontrol     15.241      6.586 5857.009   2.314 0.020693 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) blck.L blck.Q blck.C blck^4 blck^5 grpcnt blc.L: bl
c.Q:
## block.L      0.001                                                    

## block.Q      0.000  0.002                                             

## block.C     -0.001 -0.003  0.011                                      

## block^4      0.000 -0.019 -0.001  0.032                               

## block^5     -0.001 -0.001  0.003  0.000  0.019                        

## groupcontrl -0.707 -0.001  0.000  0.001  0.000  0.000                 

## blck.L:grpc -0.001 -0.687 -0.001  0.002  0.013  0.001  0.000          

## blck.Q:grpc  0.000 -0.001 -0.690 -0.007  0.001 -0.002  0.001 -0.026   

## blck.C:grpc  0.001  0.002 -0.007 -0.697 -0.022  0.000 -0.001  0.026 -0.
013
## blck^4:grpc  0.000  0.013  0.001 -0.022 -0.702 -0.014  0.000 -0.023  0.
012
## blck^5:grpc  0.000  0.001 -0.002  0.000 -0.014 -0.703 -0.001  0.004  0.
004
##             blc.C: blc^4:
## block.L                   
## block.Q                   
## block.C                   
## block^4                   
## block^5                   
## groupcontrl               
## blck.L:grpc               
## blck.Q:grpc               
## blck.C:grpc               
```

```
## blck^4:grpc  0.017
## blck^5:grpc  0.002  0.008

# training Response Time plot
ae.m.df_train1.1 = allEffects(m.df_train1.1)
ae.m.df.df_train1.1 = as.data.frame(ae.m.df_train1.1[[1]])

ae.Trainmean=ggplot(ae.m.df.df_train1.1, aes(x=block,y=fit,color=group))+
  geom_errorbar(aes(ymin=fit-se, ymax=fit+se), width=.1) +
  geom_line() +
  geom_point()+
  ylab("Response Time (ms)")+
  xlab("Block")+
  labs(color = "Group")+
  scale_color_manual(labels = c("Progressive", "Whole"),
                     values = c("#F8766D", "#00BFC4"))+
  ggtitle("Response Time during Learning")+
  theme_classic()

plot(ae.Trainmean)

## geom_path: Each group consists of only one observation. Do you need to
adjust
## the group aesthetic?
```

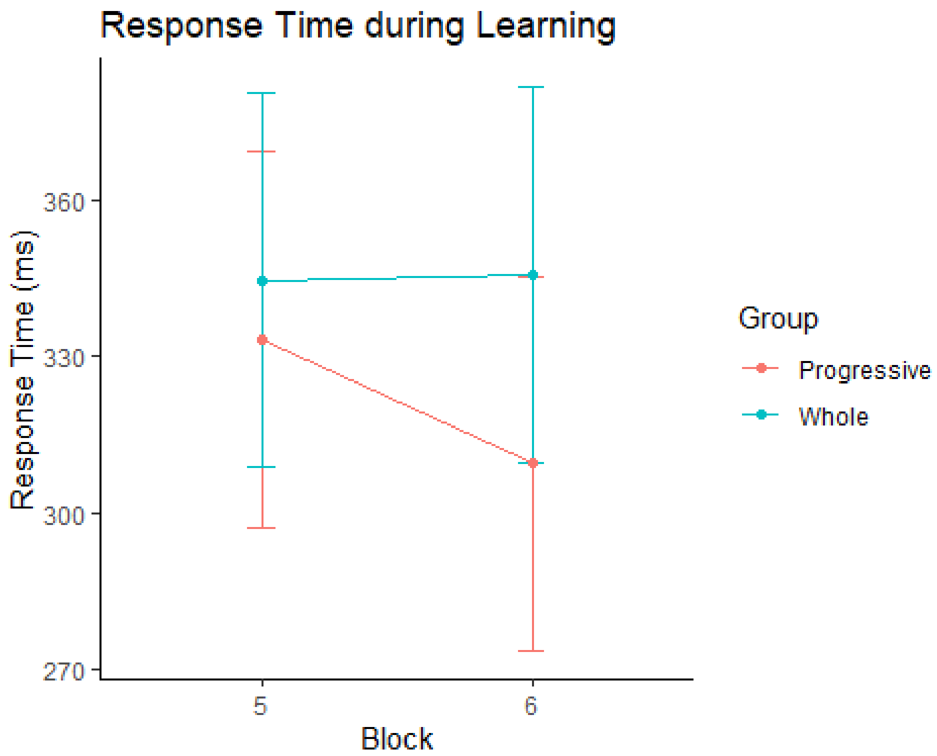


Response Time Model for Training Dataset - only blocks 5 and 6

```
# factors
df_train1.5.6$subject = factor(df_train1.5.6$subject)
df_train1.5.6$group = factor(df_train1.5.6$group)
df_train1.5.6$block = factor(df_train1.5.6$block, ordered = TRUE, levels=c
('1', '2', '3', '4', '5', '6'))
```

```
# model training
m.df_train1.5.6 = lmer(meanRT ~ block * group + (1|subject), data=df_train
1.5.6, REML = FALSE)
Anova(m.df_train1.5.6)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: meanRT
##               Chisq Df Pr(>Chisq)
## block       12.8597  1  0.0003357 ***
## group        0.2208  1  0.6384261
## block:group 15.1371  1  9.998e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m.df_train1.5.6, ddf="Satterthwaite")

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwait
e's
##   method [lmerModLmerTest]
## Formula: meanRT ~ block * group + (1 | subject)
##    Data: df_train1.5.6
##
##      AIC      BIC   logLik deviance df.resid
##  22127.8  22161.2 -11057.9  22115.8     1931
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5983 -0.5361 -0.0643  0.4519  7.1459
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 15393    124.1
##  Residual              4971     70.5
## Number of obs: 1937, groups:  subject, 24
##
## Fixed effects:
##                      Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)           321.319     35.887   23.989   8.954 4.08e-09 ***
## block.L               -16.858      3.189 1913.011  -5.286 1.40e-07 ***
## groupcontrol           23.781     50.754   23.992   0.469 0.643619
## block.L:groupcontrol   17.648      4.536 1913.023   3.891 0.000103 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) blck.L grpcnt
## block.L     -0.002
## groupcontrl -0.707  0.001
## blck.L:grpc  0.001 -0.703  0.000

# training Response Time (blocks 5 and 6) plot
ae.m.df_train1.5.6 = allEffects(m.df_train1.5.6)
```

```
ae.m.df.df_train1.5.6 = as.data.frame(ae.m.df_train1.5.6[[1]])

ae.Trainmean56=ggplot(ae.m.df.df_train1.5.6, aes(x=block,y=fit,color=grou
p, group=group))+
  geom_errorbar(aes(ymin=fit-se, ymax=fit+se), width=.1) +
  geom_line() +
  geom_point()+
  ylab("Response Time (ms)")+
  xlab("Block")+
  labs(color = "Group")+
  scale_color_manual(labels = c("Progressive", "Whole"),
                     values = c("#F8766D", "#00BFC4"))+
  ggtitle("Response Time during Learning")+
  theme_classic()

plot(ae.Trainmean56)
```



Prepare Testing Dataset for Response Time Model

```
## calculate the mean value of the block-means

# variables
test_subjects = vector()
test_blocks = vector()
test_means = vector()
test_SD = vector()

# create block means
for (subject in unique(df_testacc$subject)) {
  for (block in 7:8) {
    current_block_means =
```

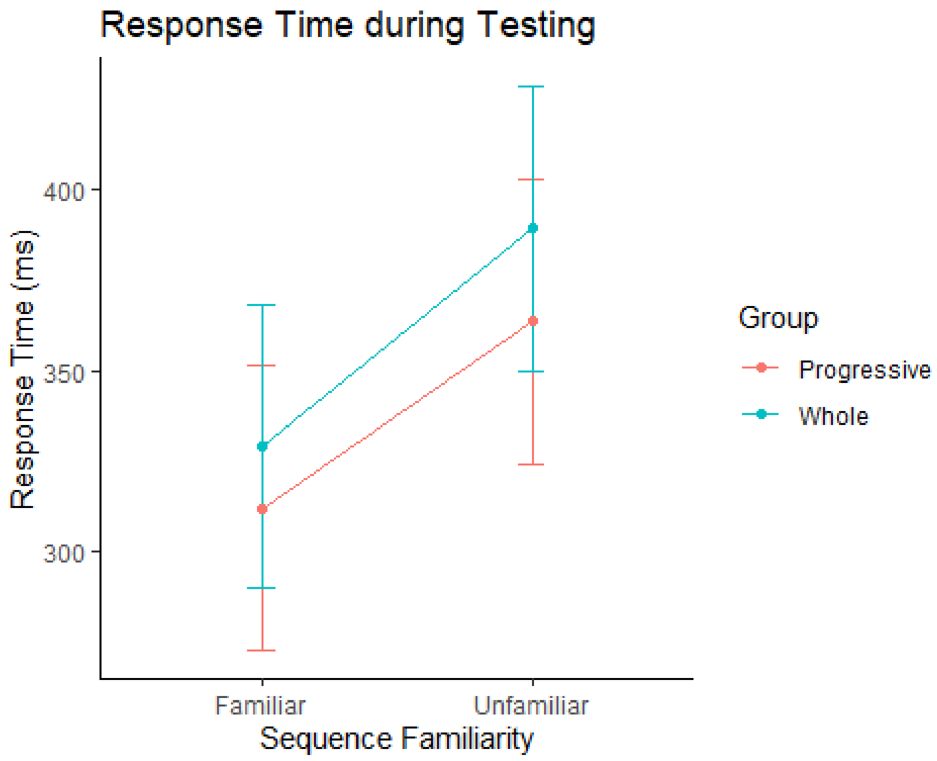
```
      df_testacc$meanRT[df_testacc$subject==subject
                        & df_testacc$block==block]

    test_subjects = append(test_subjects, subject)
    test_blocks = append(test_blocks, block)
    test_means = append(test_means, mean(current_block_means))
    test_SD = append(test_SD, sd(current_block_means))
  }
}

# create new df with the block means
df_means_test = data.frame(subject=test_subjects,
                           block=test_blocks,
                           block_mean=test_means,
                           block_sd=test_SD)

# filter accurate trials which are no more than 2.5SDs away from the mean
df_testacc2 = merge(x=df_testacc, y=df_means_test, by=c("subject", "block
"), sort=FALSE)

df_test1 = df_testacc2 %>% filter(meanRT < (block_mean+2.5*block_sd))

# retained trials: 79.73%
```

Response Time Model for Testing Dataset

```
# factors
df_test1$subject = factor(df_test1$subject)
df_test1$group = factor(df_test1$group)
df_test1$FAM_UNFAM = factor(df_test1$FAM_UNFAM)

# model testing
m.df_test1.1 = lmer(meanRT ~ FAM_UNFAM * group + (1|subject), data=df_test
1, REML = FALSE)
Anova(m.df_test1.1)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: meanRT
##                    Chisq Df Pr(>Chisq)
## FAM_UNFAM       194.6737  1     <2e-16 ***
## group             0.1453  1     0.7031
## FAM_UNFAM:group   1.1271  1     0.2884
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m.df_test1.1, ddf="Satterthwaite")

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwait
e's
##   method [lmerModLmerTest]
## Formula: meanRT ~ FAM_UNFAM * group + (1 | subject)
##    Data: df_test1
##
##      AIC      BIC   logLik deviance df.resid
```

```
##  21700.5  21733.6 -10844.3  21688.5     1831
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.6344 -0.6013 -0.1101  0.4784  7.5857
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 18477    135.93
##  Residual              7331     85.62
## Number of obs: 1837, groups:  subject, 24
##
## Fixed effects:
##                              Estimate Std. Error       df t value Pr(>|t
|)
## (Intercept)                   311.827     39.430   24.206   7.908 3.64e-
08 ***
## FAM_UNFAMUNFAM                 51.844      5.682 1813.209   9.125  < 2e-
16 ***
## groupcontrol                   17.173     55.769   24.219   0.308    0.7
61
## FAM_UNFAMUNFAM:groupcontrol    8.538      8.042 1813.217   1.062    0.2
89
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                 (Intr) FAM_UNFAMUNFAM grpcnt
## FAM_UNFAMUNFAM  -0.066
## groupcontrl     -0.707  0.047
## FAM_UNFAMUNFAM:  0.047 -0.706         -0.068
```

```
# testing Response Time plot
ae.m.df_test1.1 = allEffects(m.df_test1.1)
ae.m.df.df_test1.1 = as.data.frame(ae.m.df_test1.1[[1]])

ae.Testmean=ggplot(ae.m.df.df_test1.1, aes(x=FAM_UNFAM,y=fit,color=group,
group=group))+
  geom_errorbar(aes(ymin=fit-se, ymax=fit+se), width=.1) +
  geom_line() +
  geom_point()+
  ylab("Response Time (ms)")+
  scale_x_discrete("Sequence Familiarity", labels = c("FAM" = "Familiar",
"UNFAM" = "Unfamiliar"))+
  labs(color = "Group")+
  scale_color_manual(labels = c("Progressive", "Whole"),
                     values = c("#F8766D", "#00BFC4"))+
  ggtitle("Response Time during Testing")+
  theme_classic()

plot(ae.Testmean)
```



Prepare Single Step Dataset for Concatenation Model

```
# import dataset
dfbiggest = read_excel("C:/Users/liave/Documents/Module 12/R/ChunkingContr
olMergedLia.xlsx")

# only first 6 blocks
dfbigfirst6blocks = dfbiggest %>% subset(Session <7)
dfbigfirst6blocks = dfbigfirst6blocks %>% subset(Session >4)
# only single steps
dfbigonlysteps = dfbigfirst6blocks %>% subset (SubTrial <14)

# accuracy
dfbig = subset(dfbigonlysteps, feedback.ACC.trial=="1")

# for later analyses
dfstep1.3 = dfbig %>% subset(step.number <4)
dfstep1.3.5 = dfstep1.3 %>% subset(Session ==5)
dfstep1.3.6 = dfstep1.3 %>% subset(Session ==6)

dfstep4.6 = dfbig %>% subset(step.number >3)
dfstep4.6.5 = dfstep4.6 %>% subset(Session ==5)
dfstep4.6.6 = dfstep4.6 %>% subset(Session ==6)
```

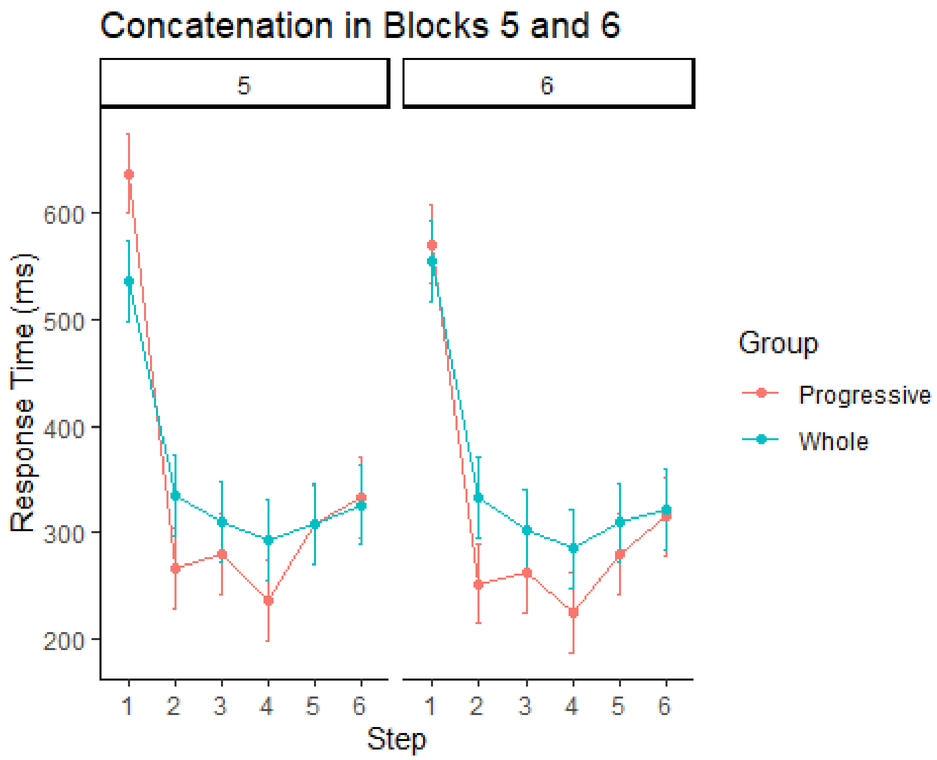
Overall Concatenation Model

```
# factors
dfbig$Subject = factor(dfbig$Subject)
dfbig$Group = factor(dfbig$Group)
dfbig$Session = factor(dfbig$Session)
dfbig$step.number = factor(dfbig$step.number)
```

```
# concatenation model
m.dfbig = lmer(feedback.RT ~ Group * Session * step.number + (1|Subject),
data=dfbig, REML = FALSE)
Anova(m.dfbig)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: feedback.RT
##                                Chisq Df Pr(>Chisq)
## Group                         0.1612  1  0.6880632
## Session                      11.8966  1  0.0005624 ***
## step.number                2978.1917  5  < 2.2e-16 ***
## Group:Session                10.6233  1  0.0011167 **
## Group:step.number           124.4015  5  < 2.2e-16 ***
## Session:step.number           1.8303  5  0.8720872
## Group:Session:step.number    12.5581  5  0.0278910 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m.dfbig, ddf="Satterthwaite")

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwait
e's
##   method [lmerModLmerTest]
## Formula: feedback.RT ~ Group * Session * step.number + (1 | Subject)
##    Data: dfbig
##
##      AIC      BIC   logLik deviance df.resid
## 161394.7 161586.7 -80671.4 161342.7    11872
##
## Scaled residuals:
##    Min     1Q Median     3Q    Max
## -3.466 -0.407 -0.078  0.246 32.085
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 15801    125.7
##  Residual             44901    211.9
## Number of obs: 11898, groups:  Subject, 24
##
## Fixed effects:
##                                      Estimate Std. Error       df t value
## (Intercept)                            636.29      37.53    27.12  16.955
## Groupcontrol                          -101.35      53.07    27.12  -1.910
## Session6                               -66.18      13.36 11874.02  -4.954
## step.number2                          -369.63      13.52 11873.99 -27.331
## step.number3                          -355.96      13.52 11873.99 -26.321
## step.number4                          -399.14      13.52 11873.99 -29.514
## step.number5                          -328.85      13.52 11873.99 -24.316
## step.number6                          -303.14      13.52 11873.99 -22.415
## Groupcontrol:Session6                   84.79      19.04 11874.04   4.453
## Groupcontrol:step.number2              169.77      19.11 11873.99   8.886
## Groupcontrol:step.number3              131.81      19.11 11873.99   6.899
## Groupcontrol:step.number4              156.86      19.11 11873.99   8.210
```

```
## Groupcontrol:step.number5            102.60      19.11 11873.99   5.370
## Groupcontrol:step.number6             94.49      19.11 11873.99   4.945
## Session6:step.number2                 52.06      18.89 11873.99   2.755
## Session6:step.number3                 48.75      18.89 11873.99   2.580
## Session6:step.number4                 53.73      18.89 11873.99   2.844
## Session6:step.number5                 38.86      18.89 11873.99   2.057
## Session6:step.number6                 48.34      18.89 11873.99   2.558
## Groupcontrol:Session6:step.number2   -72.28      26.93 11873.99  -2.684
## Groupcontrol:Session6:step.number3   -75.73      26.93 11873.99  -2.813
## Groupcontrol:Session6:step.number4   -80.10      26.93 11873.99  -2.975
## Groupcontrol:Session6:step.number5   -56.63      26.93 11873.99  -2.103
## Groupcontrol:Session6:step.number6   -71.85      26.93 11873.99  -2.668
##                                    Pr(>|t|)
## (Intercept)                        5.82e-16 ***
## Groupcontrol                        0.06679 .
## Session6                           7.39e-07 ***
## step.number2                        < 2e-16 ***
## step.number3                        < 2e-16 ***
## step.number4                        < 2e-16 ***
## step.number5                        < 2e-16 ***
## step.number6                        < 2e-16 ***
## Groupcontrol:Session6              8.55e-06 ***
## Groupcontrol:step.number2           < 2e-16 ***
## Groupcontrol:step.number3          5.51e-12 ***
## Groupcontrol:step.number4          2.44e-16 ***
## Groupcontrol:step.number5          8.02e-08 ***
## Groupcontrol:step.number6          7.70e-07 ***
## Session6:step.number2               0.00587 **
## Session6:step.number3               0.00989 **
## Session6:step.number4               0.00446 **
## Session6:step.number5               0.03970 *
## Session6:step.number6               0.01053 *
## Groupcontrol:Session6:step.number2  0.00727 **
## Groupcontrol:Session6:step.number3  0.00492 **
## Groupcontrol:Session6:step.number4  0.00294 **
## Groupcontrol:Session6:step.number5  0.03547 *
## Groupcontrol:Session6:step.number6  0.00763 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 24 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it

# plot concatenation
ae.m.dfbig = allEffects(m.dfbig)
ae.m.df.dfbig = as.data.frame(ae.m.dfbig[[1]])

ae.3factors = ggplot(ae.m.df.dfbig,
aes(x=step.number, y=fit, color=Group, group=Group))+
  geom_errorbar(aes(ymin=fit-se, ymax=fit+se), width=.1) +
  geom_line() +
  geom_point()+
```

```
  ylab("Response Time (ms)")+
  xlab("Step")+
  scale_color_manual(labels = c("Progressive", "Whole"),
                     values = c("#F8766D", "#00BFC4"))+
  ggtitle("Concatenation in Blocks 5 and 6")+
  facet_grid(~Session)+
  theme_classic()

plot(ae.3factors)
```

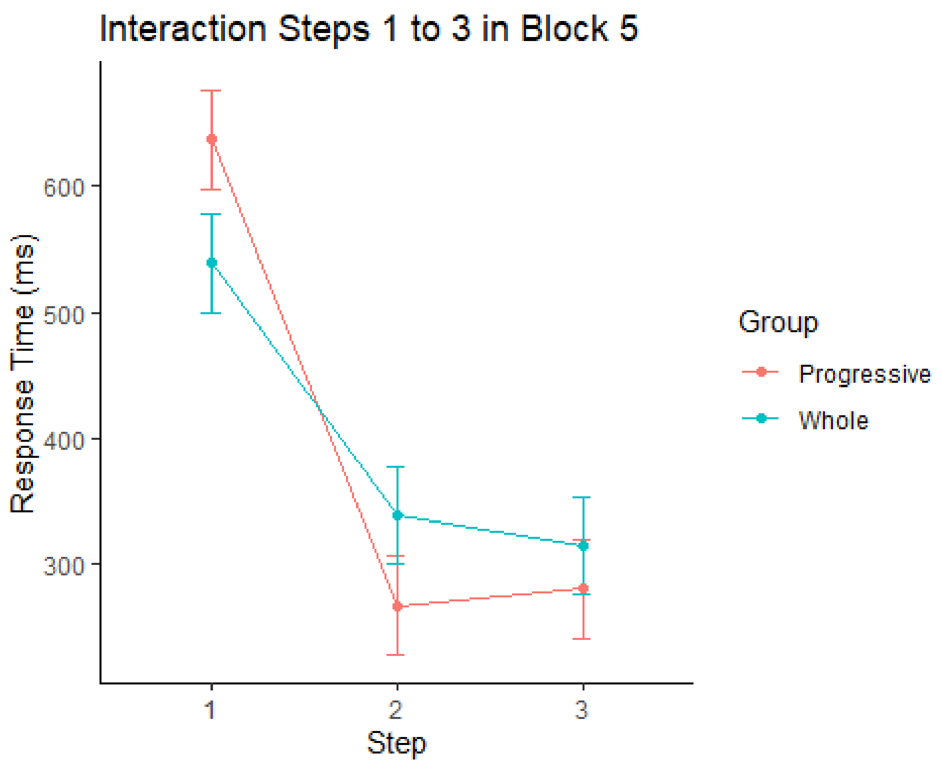


Interaction Between Steps 1 to 3 in Block 5

```
# factors
dfstep1.3.5$Subject = factor(dfstep1.3.5$Subject)
dfstep1.3.5$Group = factor(dfstep1.3.5$Group)
dfstep1.3.5$step.number = factor(dfstep1.3.5$step.number)

# model
m.dfstep1.3.5 = lmer(feedback.RT ~ Group * step.number + (1|Subject), data
=dfstep1.3.5, REML = FALSE)
Anova(m.dfstep1.3.5)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: feedback.RT
##                        Chisq Df Pr(>Chisq)
## Group                 0.0022  1     0.9625
## step.number        1187.0446  2     <2e-16 ***
## Group:step.number    85.6463  2     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(m.dfstep1.3.5, ddf="Satterthwaite")

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwait
e's
##   method [lmerModLmerTest]
## Formula: feedback.RT ~ Group * step.number + (1 | Subject)
##    Data: dfstep1.3.5
##
##      AIC      BIC   logLik deviance df.resid
##  40154.5  40202.4 -20069.2  40138.5     2944
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9604 -0.4793 -0.0644  0.3708 13.0635
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 17172    131.0
##  Residual             45610    213.6
## Number of obs: 2952, groups:  Subject, 24
##
## Fixed effects:
##                            Estimate Std. Error      df t value Pr(>|t|)

## (Intercept)                  636.82      39.04   25.99  16.311 3.59e-15
***
## Groupcontrol                 -97.99      55.21   25.99  -1.775   0.0877
 .
## step.number2                -369.63      13.63 2927.93 -27.118  < 2e-16
***
## step.number3                -355.96      13.63 2927.93 -26.116  < 2e-16
***
## Groupcontrol:step.number2   169.77      19.26 2927.93   8.816  < 2e-16
***
## Groupcontrol:step.number3   131.81      19.26 2927.93   6.845 9.28e-12
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Grpcnt stp.n2 stp.n3 Grp:.2
## Groupcontrl -0.707
## step.numbr2 -0.175  0.123
## step.numbr3 -0.175  0.123  0.500
## Grpcntrl:.2  0.124 -0.174 -0.708 -0.354
## Grpcntrl:.3  0.124 -0.174 -0.354 -0.708  0.500

# plot interaction steps 1 to 3 block 5
ae.m.dfstep1.3.5 = allEffects(m.dfstep1.3.5)
ae.m.df.dfstep1.3.5 = as.data.frame(ae.m.dfstep1.3.5[[1]])

ae.steps13.5 = ggplot(ae.m.df.dfstep1.3.5,
aes(x=step.number, y=fit, color=Group, group=Group))+
  geom_errorbar(aes(ymin=fit-se, ymax=fit+se), width=.1) +
```

```
  geom_line() +
  geom_point()+
  ylab("Response Time (ms)")+
  xlab("Step")+
  scale_color_manual(labels = c("Progressive", "Whole"),
                     values = c("#F8766D", "#00BFC4"))+
  ggtitle("Interaction Steps 1 to 3 in Block 5")+
  theme_classic()

plot(ae.steps13.5)
```

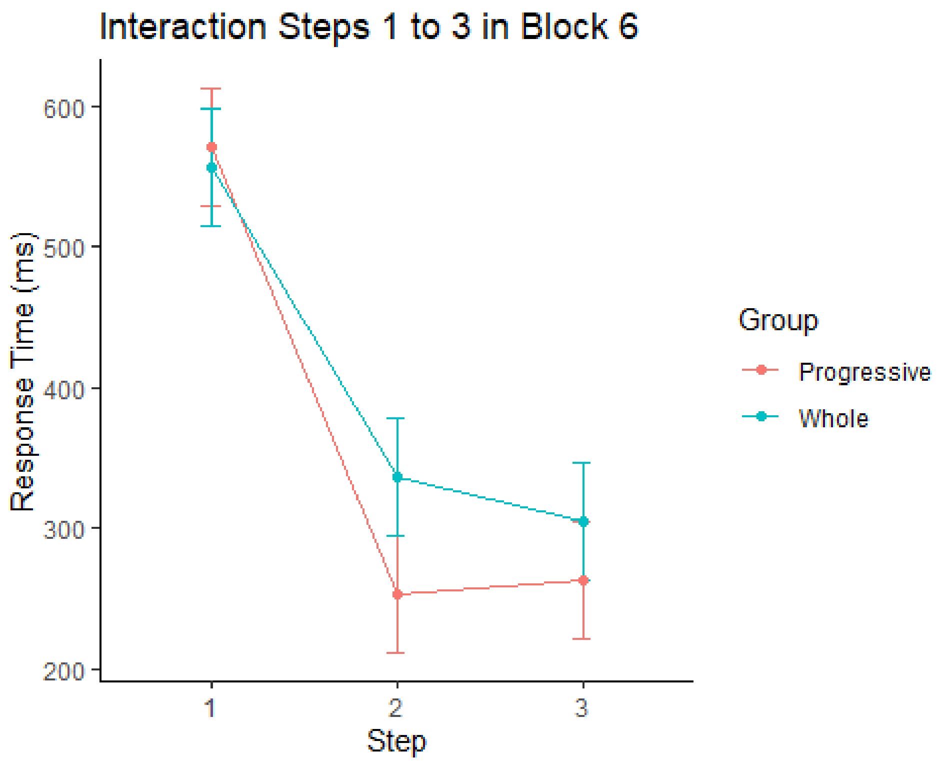


Interaction Between Steps 1 to 3 in Block 6

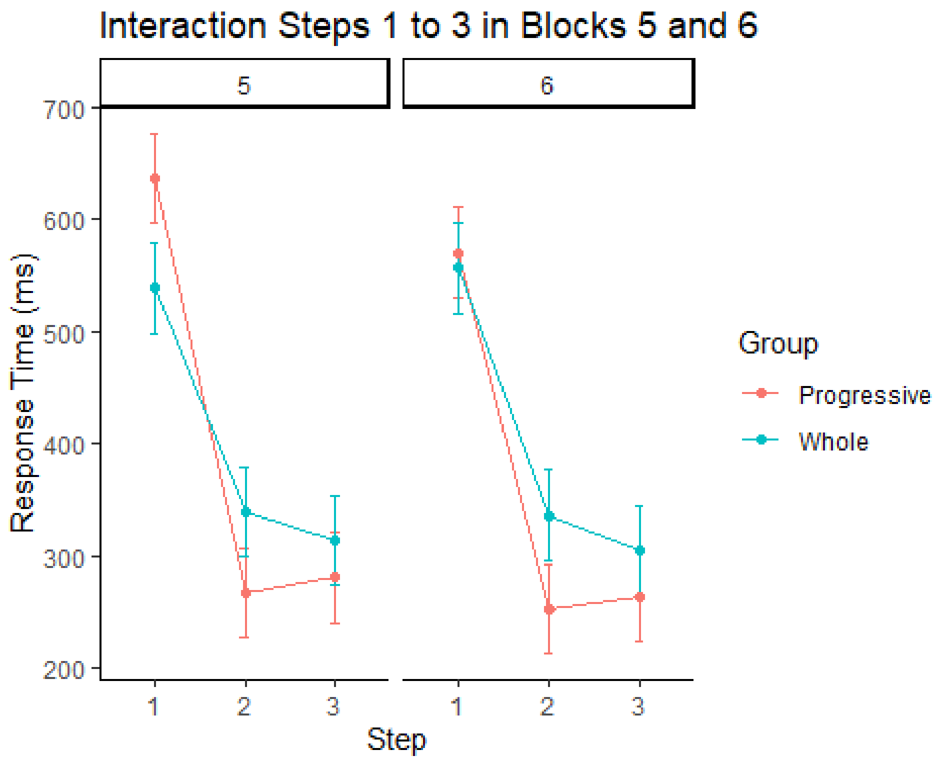
```
# factors
dfstep1.3.6$Subject = factor(dfstep1.3.6$Subject)
dfstep1.3.6$Group = factor(dfstep1.3.6$Group)
dfstep1.3.6$step.number = factor(dfstep1.3.6$step.number)

# model
m.dfstep1.3.6 = lmer(feedback.RT ~ Group * step.number + (1|Subject), data
=dfstep1.3.6, REML = FALSE)
Anova(m.dfstep1.3.6)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: feedback.RT
##                      Chisq Df Pr(>Chisq)
## Group               0.4093  1     0.5223
## step.number       901.9505  2  < 2.2e-16 ***
## Group:step.number  21.3296  2  2.335e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m.dfstep1.3.6, ddf="Satterthwaite")

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwait
e's
##   method [lmerModLmerTest]
## Formula: feedback.RT ~ Group * step.number + (1 | Subject)
##    Data: dfstep1.3.6
##
##      AIC      BIC   logLik deviance df.resid
##  41378.7  41426.7 -20681.3  41362.7     2989
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9420 -0.4824 -0.0590  0.3295 13.7037
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 19505    139.7
##  Residual             56003    236.6
## Number of obs: 2997, groups:  Subject, 24
##
## Fixed effects:
##                            Estimate Std. Error      df t value Pr(>|t|)

## (Intercept)                  570.80      41.64   25.98  13.707 2.11e-13
***
## Groupcontrol                 -14.28      58.97   26.11  -0.242  0.81048

## step.number2                -317.57      14.73 2972.91 -21.555  < 2e-16
***
## step.number3                -307.22      14.73 2972.91 -20.852  < 2e-16
***
## Groupcontrol:step.number2    97.49      21.19 2972.91   4.601 4.38e-06
***
## Groupcontrol:step.number3    56.08      21.19 2972.91   2.647  0.00818
**
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) Grpcnt stp.n2 stp.n3 Grp:.2
## Groupcontrl -0.706
## step.numbr2 -0.177  0.125
## step.numbr3 -0.177  0.125  0.500
## Grpcntrl:.2  0.123 -0.180 -0.695 -0.348
## Grpcntrl:.3  0.123 -0.180 -0.348 -0.695  0.500

# plot interaction steps 1 to 3 block 6
ae.m.dfstep1.3.6 = allEffects(m.dfstep1.3.6)
ae.m.df.dfstep1.3.6 = as.data.frame(ae.m.dfstep1.3.6[[1]])
```

```
ae.steps13.6 = ggplot(ae.m.df.dfstep1.3.6,
aes(x=step.number, y=fit, color=Group, group=Group))+
  geom_errorbar(aes(ymin=fit-se, ymax=fit+se), width=.1) +
  geom_line() +
  geom_point()+
  ylab("Response Time (ms)")+
  xlab("Step")+
  scale_color_manual(labels = c("Progressive", "Whole"),
                     values = c("#F8766D", "#00BFC4"))+
  ggtitle("Interaction Steps 1 to 3 in Block 6")+
  theme_classic()

plot(ae.steps13.6)
```



Plotting Steps 1 to 3 - blocks 5 and 6 together

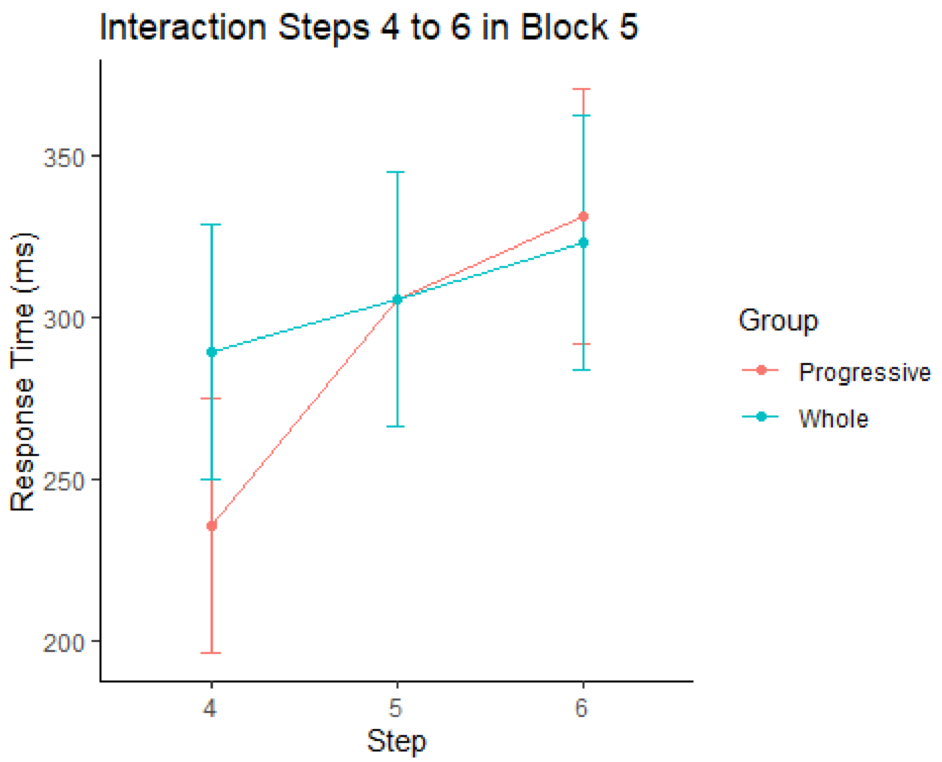


Interaction Between Steps 4 to 6 in Block 5

```
# factors
dfstep4.6.5$Subject = factor(dfstep4.6.5$Subject)
dfstep4.6.5$Group = factor(dfstep4.6.5$Group)
dfstep4.6.5$Session = factor(dfstep4.6.5$Session)
dfstep4.6.5$step.number = factor(dfstep4.6.5$step.number)

# model
m.dfstep4.6.5 = lmer(feedback.RT ~ Group * step.number + (1|Subject), data
=dfstep4.6.5, REML = FALSE)
Anova(m.dfstep4.6.5)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: feedback.RT
##                      Chisq Df Pr(>Chisq)
## Group               0.0778  1   0.780304
## step.number        47.1245  2  5.849e-11 ***
## Group:step.number  12.4712  2   0.001958 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m.dfstep4.6.5, ddf="Satterthwaite")

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwait
e's
##   method [lmerModLmerTest]
## Formula: feedback.RT ~ Group * step.number + (1 | Subject)
##    Data: dfstep4.6.5
```

```
##
##      AIC      BIC   logLik deviance df.resid
##  40139.6  40187.5 -20061.8  40123.6     2944
##
## Scaled residuals:
##    Min      1Q Median      3Q    Max
## -1.646 -0.352 -0.099  0.151 31.722
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 17391    131.9
##  Residual             45373    213.0
## Number of obs: 2952, groups:  Subject, 24
##
## Fixed effects:
##                            Estimate Std. Error      df t value Pr(>|t|)

## (Intercept)                  235.63      39.27   26.02   6.000 2.45e-06
***
## Groupcontrol                  54.06      55.53   26.02   0.973  0.33932

## step.number5                  70.29      13.59 2927.98   5.170 2.49e-07
***
## step.number6                  96.00      13.59 2927.98   7.061 2.05e-12
***
## Groupcontrol:step.number5    -54.26      19.21 2927.98  -2.825  0.00476
**
## Groupcontrol:step.number6    -62.38      19.21 2927.98  -3.248  0.00118
**
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) Grpcnt stp.n5 stp.n6 Grp:.5
## Groupcontrl -0.707
## step.numbr5 -0.173  0.122
## step.numbr6 -0.173  0.122  0.500
## Grpcntrl:.5  0.123 -0.173 -0.708 -0.354
## Grpcntrl:.6  0.123 -0.173 -0.354 -0.708  0.500

# plot interaction steps 4 to 6 in block 5
ae.m.dfstep4.6.5 = allEffects(m.dfstep4.6.5)
ae.m.df.dfstep4.6.5 = as.data.frame(ae.m.dfstep4.6.5[[1]])

ae.steps46.5 = ggplot(ae.m.df.dfstep4.6.5,
aes(x=step.number, y=fit, color=Group, group=Group))+
  geom_errorbar(aes(ymin=fit-se, ymax=fit+se), width=.1) +
  geom_line() +
  geom_point()+
  ylab("Response Time (ms)")+
  xlab("Step")+
  scale_color_manual(labels = c("Progressive", "Whole"),
                     values = c("#F8766D", "#00BFC4"))+
  ggtitle("Interaction Steps 4 to 6 in Block 5")+
```

```
  theme_classic()

plot(ae.steps46.5)
```

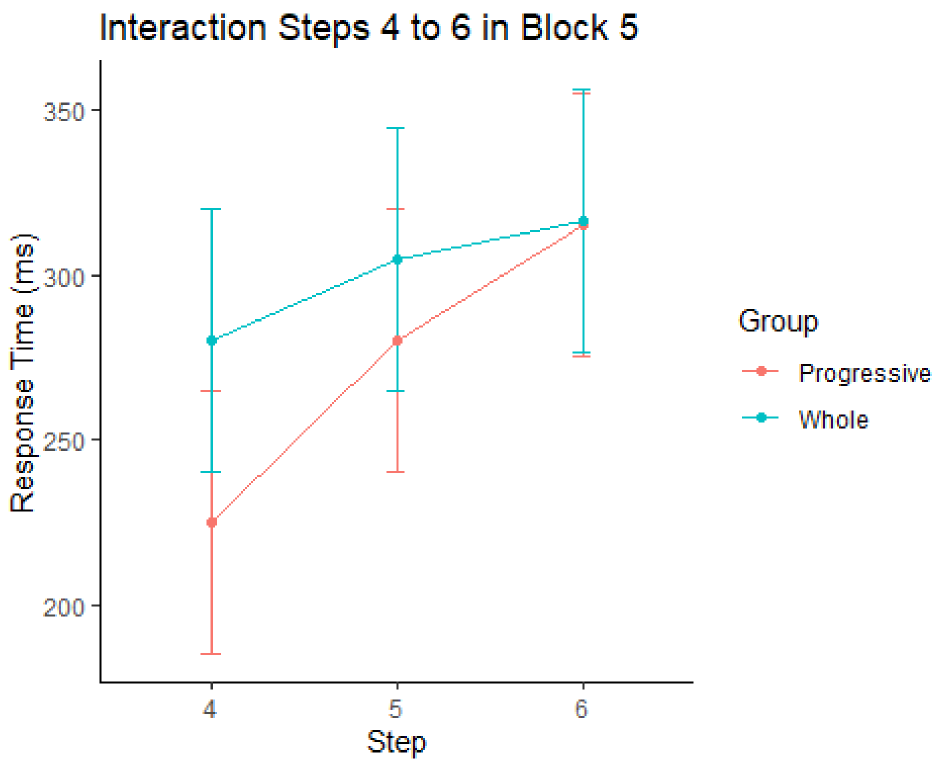


Interaction Between Steps 4 to 6 in Block 6

```
# factors
dfstep4.6.6$Subject = factor(dfstep4.6.6$Subject)
dfstep4.6.6$Group = factor(dfstep4.6.6$Group)
dfstep4.6.6$Session = factor(dfstep4.6.6$Session)
dfstep4.6.6$step.number = factor(dfstep4.6.6$step.number)

# model
m.dfstep4.6.6 = lmer(feedback.RT ~ Group * step.number + (1|Subject), data
=dfstep4.6.6, REML = FALSE)
Anova(m.dfstep4.6.6)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: feedback.RT
##                      Chisq Df Pr(>Chisq)
## Group               0.2316  1  0.6303732
## step.number        92.0965  2  < 2.2e-16 ***
## Group:step.number 15.9700  2  0.0003405 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(m.dfstep4.6.6, ddf="Satterthwaite")

## Linear mixed model fit by maximum likelihood . t-tests use Satterthwait
e's
##   method [lmerModLmerTest]
```

```
## Formula: feedback.RT ~ Group * step.number + (1 | Subject)
##    Data: dfstep4.6.6
##
##      AIC      BIC   logLik deviance df.resid
##  38734.3  38782.3 -19359.1  38718.3     2989
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.2558 -0.4294 -0.1002  0.2092 15.2647
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 18508    136.0
##  Residual             23022    151.7
## Number of obs: 2997, groups:  Subject, 24
##
## Fixed effects:
##                             Estimate Std. Error       df t value Pr(>|t|)

## (Intercept)                  224.851     39.837   24.917   5.644 7.19e-06
 ***
## Groupcontrol                  55.164     56.370   24.974   0.979   0.3372

## step.number5                  55.422      9.446 2973.016   5.867 4.92e-09
 ***
## step.number6                  90.607      9.446 2973.016   9.592  < 2e-16
 ***
## Groupcontrol:step.number5    -30.787     13.585 2973.016  -2.266   0.0235
 *
## Groupcontrol:step.number6    -54.120     13.585 2973.016  -3.984 6.95e-05
 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Grpcnt stp.n5 stp.n6 Grp:.5
## Groupcontrl -0.707
## step.numbr5 -0.119  0.084
## step.numbr6 -0.119  0.084  0.500
## Grpcntrl:.5  0.082 -0.121 -0.695 -0.348
## Grpcntrl:.6  0.082 -0.121 -0.348 -0.695  0.500

# plot interaction steps 4 to 6 in block 6
ae.m.dfstep4.6.6 = allEffects(m.dfstep4.6.6)
ae.m.df.dfstep4.6.6 = as.data.frame(ae.m.dfstep4.6.6[[1]])

ae.steps46.6 = ggplot(ae.m.df.dfstep4.6.6,
aes(x=step.number, y=fit, color=Group, group=Group))+
  geom_errorbar(aes(ymin=fit-se, ymax=fit+se), width=.1) +
  geom_line() +
  geom_point()+
  ylab("Response Time (ms)")+
  xlab("Step")+
  scale_color_manual(labels = c("Progressive", "Whole"),
```

```
                        values = c("#F8766D", "#00BFC4"))+
  ggtitle("Interaction Steps 4 to 6 in Block 5")+
  theme_classic()

plot(ae.steps46.6)
```



Plotting Steps 4 to 6 - blocks 5 and 6 together