The scaling of mental computation in a sorting task

Susanne Haridi
Max Planck Institute for Biological Cybernetics
Max Planck School of Cognition

Charley M. Wu
University of Tübingen

Ishita Dasgupta
DeepMind New York

Eric Schulz
Max Planck Institute for Biological Cybernetics

Many cognitive models provide valuable insights into human behavior. Yet the algorithmic complexity of candidate models can fail to capture how human reaction times scale with increasing input complexity. In the current work, we want to understand the algorithms underlying human cognitive processes. Computer science characterizes algorithms by their time and space complexity scaling with problem size. We propose to use participants' reaction times to study how human computations scale with increasing input complexity. We test this approach in a task where participants had to sort sequences of rectangles by their size. Our results showed that reaction times scaled linearly with sequence length and that participants learned and actively used latent structure whenever it was provided. This behavior was in line with a computational model that used the observed sequences to form hypotheses about the latent structures, searching through candidate hypotheses in a directed fashion. These results enrich our understanding of plausible cognitive models for efficient mental sorting and pave the way for future studies using reaction times to investigate the scaling of mental computations across psychological domains.

Keywords: Mental Sorting, Complexity, Visual Search, Structure Learning, Reaction Times

Introduction

Imagine you are in a supermarket. Normally, choosing a box of cereal takes you around one minute. However, today the selection of cereal brands has expanded from 4 to 20. What does that mean for the time it will take you to make up your mind?

In daily life, people are faced with a plethora of tasks that vary in scope and complexity. For many of these tasks (like choosing between 4 boxes or 20 boxes of cereal), humans cope well with arbitrary changes in complexity or size. Yet there is still much we do not know about how the dimensions of a task or the size of the inputs affect the complexity of cognitive computations in humans. Moreover, many cognitive models, lack the scalability that everyday human behavior seems to suggest.

An example of how unrealistic this scaling can

Correspondence should be addressed to Susanne Haridi, Max Planck Institute for Biological Cybernetics, Germany. Email: susanne.haridi@maxplanckschools.de

be is Gaussian process regression, which has been used to describe human function learning (Lucas, Griffiths, Williams, & Kalish, 2015; Schulz, Tenenbaum, Duvenaud, Speekenbrink, & Gershman, 2017) and generalization (Schulz et al., 2019; Wu, Schulz, Speekenbrink, Nelson, & Meder, 2018). Computing the posterior of a Gaussian process scales cubically with the size of the input, which means increasing the input size from 4 to 20 (as in our cereal example) would transform a simple one minute-long task into a laborious two hour-long ordeal.

Since all psychological algorithms must eventually be implemented *in vivo* by bounded agents with limited time and computational capacities (Gershman, Horvitz, & Tenenbaum, 2015; Gigerenzer & Brighton, 2009; Gigerenzer & Selten, 2002; Lieder & Griffiths, 2020; Simon, 1990), the *complexity* of the proposed algorithms (Bossaerts & Murawski, 2017; Van Rooij, 2008), specifically the amount of processing time to perform a computation, is a reasonable constrain to select for plausible models.

An informative way to characterize an algorithm's complexity is to consider how the processing time and the required memory scale with the problem or input size. This is standard practice in computer science, where the complexity of an algorithm is measured by using the big O notation. This notation classifies algorithms according to how the processing time or memory required increases as the input size grows (Papadimitriou, 2003). In the current study, we focus on time complexity (i.e. the processing time), which is what we will refer to when talking about complexity. Similarly, we will use the term "scaling" to refer to how time complexity increases with input size. As a rule of thumb, constant processing time complexities are ideal, logarithmic complexities are favorable, linear complexities are tolerable, and polynomial complexities such as cubic scaling are to be avoided whenever possible. Yet many psychological models scale worse than linearly, i.e. superlinearly (Van Rooij & Wareham, 2008), as seen in the example of a Gaussian process mentioned earlier.

But how can the scaling of mental computations be investigated? Are there features of human cognition, for example, the use of latent structures, that can help improve the scaling of mental computations? And what type of models can capture this scaling? One way to approach these questions is to treat the human mind as a black box server and then use methods inspired by algorithmic complexity attacks (Crosby & Wallach, 2003): send the server problems of varying input size and track its computing time. This would allow us to estimate the algorithmic complexity of the current computations based on the relationship between input size and response time. Following this logic, we can create experiments, varying the number of input points and the underlying structure of the task. By measuring participants' reaction times (RTs), we can approximate the set of plausible algorithms underlying participants' mental computations.

In the current work, we apply this approach to a mental sorting task. Sorting paradigms have a valuable history in psychological research (Ashcraft & Battaglia, 1978; Berg, 1948; McGonigle & Chalmers, 2002), in particular in developmental psychology (Inhelder & Piaget, 1958; Young & Piaget, 1976). Earlier studies conducted by Piaget and colleagues on children and adults' seriation behavior (Young & Piaget, 1976) provided evidence for super-linear scaling in sorting. In these tasks, participants were asked to sort physical objects from the smallest to the largest element. However, in recent years there has been less research in this domain, and the question of how humans sort remains largely open. The importance of sorting lies in the resulting order. If humans organize (sort) information well, it can be retrieved more effectively. The smart organization of information is also a fundamental problem in computer science, where the need to search large corpora of information arises often. As such, the complexity of sorting algorithms has been widely-studied in this field (Cormen, Leiserson, Rivest, & Stein, 2009). It has, for example, been proven that

all exact sorting algorithms that use any pairwise comparison between items can at best achieve a complexity of $O(N \log N)$, i.e. scale super-linearly (Cormen et al., 2009). Many common sorting algorithms fall under this category. Merge-sort, for example, continuously splits the to-be-sorted array in half until it cannot be further divided. Each separate array then gets sorted and merged in sorted order with the array it was split from. Merge sort, as well as other such algorithms, could be valid candidates to describe participants' mental sorting behavior. Accordingly, if we find that human sorting has a linear or below linear complexity, we can differentiate between a variety of algorithms that are no longer plausible candidates for how humans sort. We can then focus on those algorithms, which remain plausible candidates to better understand the mechanisms of mental sorting in humans.

Given that some cognitive processes might scale poorly with the number of observation points, we believe that it is prudent for agents, biological or otherwise, to improve their scaling behavior by applying strategies that simplify the algorithmic complexity or reduce the data that needs to be processed.

One example that has been looked at is the classic mental rotation task by Shepard and Metzler (1971). In this task people had to determine whether two images showed the same object, just with a different rotation. It was found that the reaction times in this task increased linearly with the angle of rotation. However, researchers have wondered how people decide in which direction they rotate a given object (Hamrick & Griffiths, 2014), since the shortest direction crucially depends on the starting position in the image. In a series of studies, Hamrick and Griffiths (2014) showed that participants used the structure of the original image to efficiently choose a rotation direction, thus saving valuable computation time.

Additionally, Logan, Ulrich, and Lindsey (2016) argued that experienced typists use structure to predict future characters to achieve faster typing times as the complexity of the number of keys increases. The idea of using structure in the environment to speed up cognitive algorithms can be traced back to Brunswik (1952), who argued that people use cues in the environment to decide which strategy to apply, and studies by Harlow (1949) on learning-to-learn effects showed that repeated encounters of similar structures led participants to learn novel tasks much faster. The use of structure to reduce computational complexity lies at the core of boundedly rational accounts of cognition (Gigerenzer & Selten, 2002) and has been described as the *sine qua non* of human learning efficiency (Gershman et al., 2015; Griffiths, Lieder, & Goodman, 2015). If there is structure in the world that can speed up mental computations while maintaining accuracy, then intelligent agents should exploit this structure.

Goals

In line with these thoughts, the aim of this study is twofold. Firstly, We want to investigate how human sorting scales when more items need to be sorted. Secondly, we want to understand if and how people's sorting time can be reduced by the exploitation of latent structure in the task.

To investigate how mental sorting scales, we showed participants sequences of rectangles of different colors and sizes, which they had to mentally sort by their size. To measure how their sorting time scaled for different input sizes, we manipulated the number of rectangles. Furthermore, we also manipulated the presence or absence of different latent structures in the task. This allowed us to investigate whether participants were able to exploit latent structure to improve the time complexity for mental sorting. Our results showed that participants' RTs scaled approximately linearly with the number of rectangles and that they exploited the latent structures to reduce their response times. This behavior was captured by a linear sorting algorithm that uses information about the range of possible sizes of the rectangles to avoid a pair-wise comparison sort. Furthermore, the algorithm used the observed trials to construct hypotheses about

the underlying structure, resulting in improved efficiency. These results enrich our understanding of plausible cognitive models for efficient mental sorting and pave the way for future investigations using reaction times to probe the scaling of mental computations across psychological domains.

Methods

In order to investigate the scaling of mental computations, we studied how the time people needed to mentally sort sequences scaled when increasing the length of the sequence. Furthermore, we investigated if participants were able to detect and exploit latent structures in the presented sequences and tasks, to improve the scaling of their mental sorting.

Participants. We recruited 103 adults (37 female, age range: 24 to 74, M_{age}= 39.31; SD=11.13) via Amazon Mechanical Turk (MTurk). To ensure that the task was well-understood, all participants had to answer three comprehension questions about the task before the start of the experimental trials. Furthermore, we used a performance cutoff of 75% accuracy as an exclusion criterion. This cutoff corresponded to the average accuracy a participant could achieve if they only ever sorted the first three rectangles. Altogether, 30 participants were excluded due to performance below the cutoff (21 participants) or incomplete data (9 participants), leaving us with a sample size of 73 participants. Participants were paid up to \$11.00 (\$3.00 base fee plus a bonus of up to \$8.00; M_{reward}=\$10.3; SD=\$0.96; the bonus was linearly dependent on the accuracy, i.e. if a participant got 80% of trial correct they received a bonus of 0.8*\$8=\$6.4). The experimental task took on average about 40 Minutes (including breaks, which could be taken after each block). Informed consent was obtained from all participants. The study was approved by the ethics committee of the medical faculty of the University of Tübingen (number 701/2020BO).

Design. We used a 3×2 within-subject design to manipulate the task (*sort* vs. *memory*) and latent structure (*no structure* vs. *query structure* vs. *sequence structure*; see Fig. 1a). Additionally, we also manipulated the input size, i.e. the number of rectangles (sequence length) was varied from 1 to 7 colored rectangles of different heights in all conditions.

In the *sort task*, sequences were scrambled requiring participants to mentally sort them, while in the *memory task*, they were already sorted from the smallest (on the left) to the tallest rectangle (on the right). In both tasks, participants were asked to remember the sequences in the sorted order and then correctly report the position of a randomly queried rectangle. This meant that sequences only had to be remembered in the *memory task*, but both sorted and remembered in the *sort task*. The *memory task* was introduced to control for increases in RTs solely due to memory, allowing us to quantify the scaling of mental sorting by computing the difference in RTs between the memory and sort conditions. For this purpose, we systematically matched sequences in all trials between the two tasks by the height of the shown rectangles, the length of the sequences, and the queried position. To prevent any memory effect from one task to the other, the colors for each trial were chosen randomly from a uniform distribution over all colors (see Fig. 1A, for an example of a matched sequence for all 6 task $\times$ condition combinations). Each color only appeared once in each sequence.

To investigate the effects of latent structure on the scaling of mental sorting, we also introduced three structure conditions. Participants were not informed about the latent structures in any way, i.e. to use them they had to learn them unprompted. In the *no structure* condition, the scrambled sequences were generated randomly, meaning that colors, the height of the smallest rectangle, the position of the rectangles, and the queried position were chosen from a uniform distribution. Since all rectangles had equal differences in height to their neighbouring rectangles, the height of the smallest rectangle completely determined the height of all rectangles in a sequence. The *query structure* con-

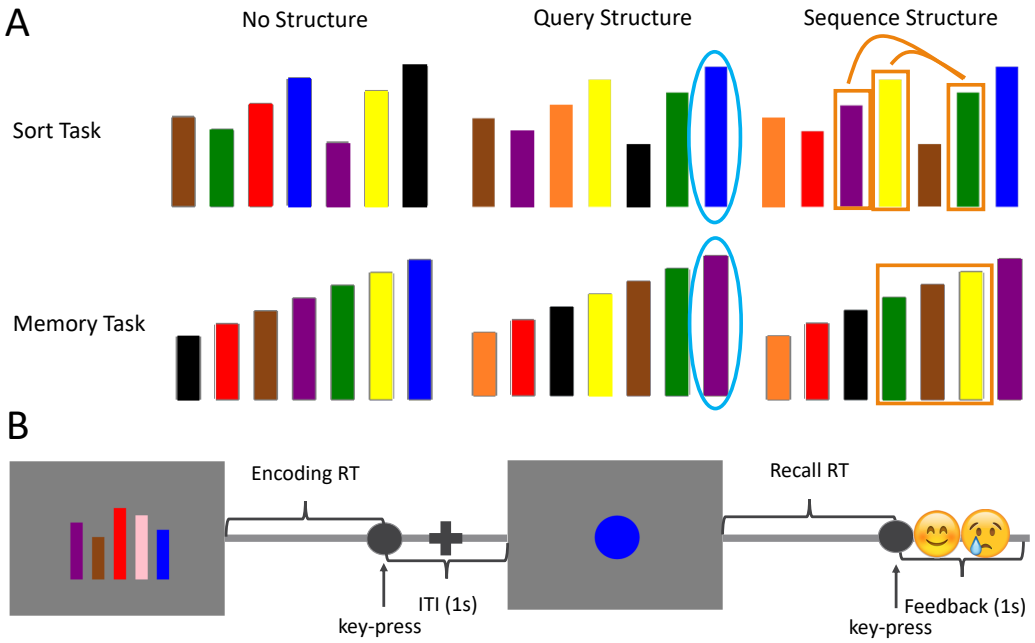

Figure 1. Overview of the experimental design. **A**) Schematic of the six different conditions. In the *no structure* condition, both the colors and the queried position were randomly sampled from a uniform distribution. In the *query structure* condition, the query always queried the tallest rectangle. In the *sequence structure* condition, three colors always followed each other in height (here purple green and yellow for the *sort task* and green, brown, and yellow for the *memory task*), though the rectangles with these colors were at random positions in the *sort task*. The sequences shown here are matched, meaning they all had the same lengths and heights and the same order for the three structure conditions in the *sort task*. Within each structure condition the same position was queried. The colors were not matched to prevent memory effects. **B**) Schematic of one trial in the *sort task*. A sequence was shown and participants used the spacebar to indicate when they had finished mentally sorting and/or memorising the sequence (encoding RT), until which it remained visible. After the presentation of a fixation cross (1s), participants were shown a colored circle (query). Participants were then asked to respond with the number key corresponding to the position of the rectangle with the same color in the sorted order ('2' in the example shown here; recall RT) Participants were given feedback about the correctness of their response. The display here left out the instructions, which were always included at the bottom of the screen to remind participants of the correct action at each stage in the trial.

dition used the same sequences as the no structure condition; however, participants were only queried about the tallest rectangle in the sequence. To prevent memory effects, the colors of the rectangles were re-sampled randomly from a uniform distribution. The latent structure of this tasks allowed participants to (theoretically) perfectly solve the task just by determining the length of the sequence. Lastly, the *sequence structure* condition also used the same sequences as the other two conditions and the queried position was randomly sampled from a uniform distribution. However, we used three reoccurring colors that were always assigned to rectangles that followed each other in height once the sequence was sorted. The two sets of three colors (one for the *sort task* and one for the *memory task* remained constant for each participant. These colors always appeared so long as the length of the

sequence allowed it. For example, if the color sequence was "purple", "green" and "yellow" (as in the example in Fig. 1A), then a sequence of length two would have a "purple" and a "green" rectangle, with the purple rectangle being the smaller one. For sequences that had more than three rectangles, the rest of the colors were sampled randomly as in other two conditions. If participants are able to learn the latent sequence structure, they should be able to connect the three rectangles into a single "entity", thus reducing the sorting time by a (theoretically) constant amount. Ror sequences of length three or below no sorting was necessary as at all.

The way we generated sequences resulted in trials which (between conditions) were matched for the length of the sequence, the heights of the rectangles and the order of the rectangles (for the *sort task*). Due to the latent structure and to prevent learning effects, the trials were not matched for queried position and colors. Each of the six combinations of the 3×2 design were presented in separate blocks. To avoid any order effects, both the order of the blocks and the order of the trials were randomized for each participant.

Materials and procedure. The experiment was conducted online and all participants were recruited via Amazon Mechanical Turk. After giving informed consent, participants were shown the instructions of the task. Specifically, they were instructed to either mentally sort (*sort task*) or to remember the pre-sorted sequences (*memory task*) as fast and accurately as possible. Participants were told that their bonus depended on the percentage of correct trials (but not the speed at which they responded). Participants were not informed about the latent structures in any way.

After the instructions, participants completed 14 *no structure* practice trials (one for each possible sequence length in randomized order) from both the *sort task* and the *memory task*. Following the practice trials, participants were required to answer three comprehension questions correctly. Afterwards, the six experimental blocks started in a fully randomized order, consisting of 35 trials (5 trials for each sequence lengths) in each block, resulting in a total number of 210 trials for each participant. At the end of the task, participants performed a short colorblindness test, and were asked to provide demographic information and an optional description about which strategies they used and whether they had noticed any differences between the blocks.

Each trial began with participants seeing a sequence of rectangles and being asked to respond by pressing the space-bar *after* they had sorted and/or memorized the sequence. We instructed participants to only press the space-bar once they had finished the sort. Accordingly, we used the time between the presentation of the sequence and the press of the space-bar (encoding RT) to measure the duration of their mental sort. As soon as they responded, the sequence disappeared and a fixation cross was shown for 1s. Afterwards, participants were shown a colored circle (query), corresponding to the color of one of the rectangles. They were then asked to respond by pressing the number key corresponding to the (sorted) position of the rectangle with the same color (see Fig. 1b for an example). In the *memory task*, this corresponded to simply remembering the position of that colored rectangles without any mental sorting. We recorded both the reaction time (RT) during which participants observed the stimuli (referred to as encoding RT from here onward) and during which they were shown the query (referred to as recall RT from here onward). As mentioned earlier, we believe that the mental sort happened during the encoding RT. Theoretically, it is also possible that participants sorted the sequence after they saw the query (recall RT). But since encoding the unsorted sequence and then sorting it form memory would require higher working memory demands than sorting the visible sequence and then remembering it in the sorted order, we believed this to be unlikely (see appendix A for further checks). Participants received feedback about the correctness of their response after every trial and at the end of each block when they

were told the percentage of correct trials for the block they had just completed.

The experiment was programmed in HTML and JavaScript with the help of the jsPsych toolbox (De Leeuw, 2015). The rectangle stimuli were generated using the psycho-physics plugin (Kuroki, 2020). The rectangles were presented at the center of the screen and were 50 pixels in width and varied in height from 150 to 390 pixels. The height difference between adjacent rectangles in the sorted order was always 30 pixels, meaning that the height of the smallest rectangle fully determined the height of all other rectangles in a sequence. To prevent uncertainty about the name of particular colors, we used colors corresponding to the 11 basic color terms (except gray, which was the background color) from the color lexicon of American English (Lindsey & Brown, 2014) for the color of our rectangles, i.e. black, white, red, yellow, green, blue, brown, orange, pink and purple.

Exclusions. We had 15330 trials in total (210 trials per participant), but for all subsequent analysis, we excluded all incorrect trials (670 trials). For the correct trials, we also excluded all trials for which either the encoding or the recall RTs were longer than 10 seconds (1216 trials), to avoid including trials, where the participant had left the screen (see Fig. C1a). This left us with 13444 trials in total.

Results

Hypotheses

We had three main hypotheses. First, we hypothesized that the encoding RT would increase with the length of the sequence, the nature of this increase (sub-linear, linear, or super-linear) being the subject of our investigation. Secondly, we hypothesized that participants would benefit from the latent structure, leading to faster encoding and better scaling. Thirdly, we hypothesized that for the encoding RT, participants would profit increasingly with increasing sequence length in the *query structure* condition, since they only ever had to identify the tallest rectangle. Similarly, we hypothesized that the encoding RTs would profit increasingly only for the first three rectangles and then remain faster by a constant amount for the *sequence structure* condition, since three rectangles always followed each other and therefore could be treated as one connected unit during mental sorting.

Behavioral results

As can be seen in Fig. 2A, the encoding RTs increased with the sequence length. To investigate which predictors are relevant for the change in the encoding RTs (see appendix A and Fig. A1 for analyses with recall RTs) we used Bayes Factors (BFs) to compare a full model with a model were the predictor (or target variable) we were investigating was excluded. Specifically, we performed model comparisons using maximally-structured mixed effects models (Barr, Levy, Scheepers, & Tily, 2013). This means that we always compared a full model containing the structure and task conditions and the number of rectangles as both random and fixed effects as well as the block number (to control for block order) as a random effect over participants against a model that did not contained the target variable as a fixed effect. If the full model is not explaining the data better than the model which misses the target variable, than the target variable is unlikely to have a strong and systematic contribution to the change in RTs. Accordingly the model comparison here only serves to confirm the relevancy of the target variables (i.e. our experimental manipulations). We used *bridge sampling* (Gronau, Singmann, & Wagenmakers, 2017) as included in the *brms* package (Bürkner, 2018; Bürkner, 2017) to approximate Bayes Factors (BF) for these comparisons. A BF that is larger than 1 provides evidence for an effect, while a BF below 1 provides evidence against it. A BF of 2 would indicate that that the data is twice as likely under the alternative hypothesis. Generally, BFs that are larger than 3 are interpreted as giving substantial evidence for one hypothesis over the other.

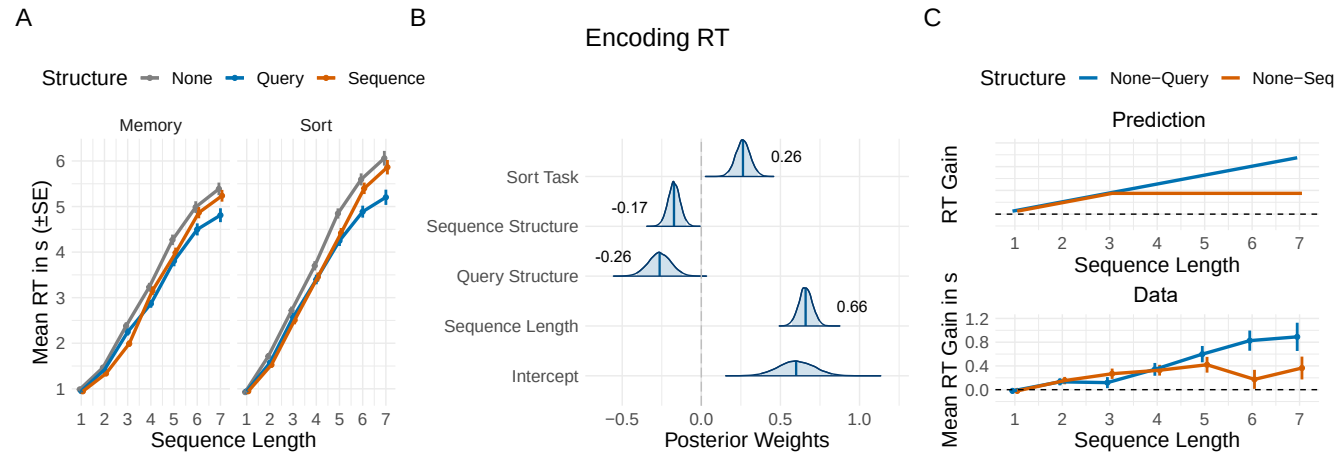

Figure 2. Behavioral results. **A**) Average encoding RT over all trials. Error-bars represent the standard error (SE). The left plot only shows the trials from the *memory task*, while the right plot shows the trials from the *sort task*. **B**) Model results of the encoding RT as the predicted variable. This plot depicts the estimates of the effects of the full model that tried to predict the encoding RTs. The model contained the displayed effects as both main and random effects and additionally also included the block number as a random effect. The depicted numbers are the estimated effects. **C**) Predicted and actual RT gain through structure. The upper plot is a schematic of the predicted gain that structure can provide if the participants were fully aware of the structure and were using it to the full extend. For the *query structure* we would expect a monotonic increase of the gain with increasing sequence length. The exact shape of this increase, however, does depend on the way that mental sorting scales. The depicted linear increase is therefore just for illustrative purposes. The lower plots shows the actual gain trough structure (calculated by taking the mean of the difference of each trial in the *no structure sort task* to the difference of the corresponding trials in the *query structure sort task* (blue) and the *sequence structure sort task* (orange). The error-bars depict the SE.

As has been done e.g. by Bartsch and Oberauer (2021), we estimated the models via an MCMC algorithm that used sampled parameter values that are proportional to the product of of the likelihood and the prior to estimate the posterior. We generated these samples with 4 independent Markov chains with 5000 warm-up samples each, followed by 5000 samples drawn from the posterior distribution. We also visually inspected the chains for convergence. All Rhat values were equal to 1.

Sequence length increases RTs, while latent structures reduce RTs. Our analysis of encoding RTs showed that the full model performed better than the model without the *structure* conditions ($BF > 100$), the number of rectangles ($BF > 100$), or the *task* conditions ($BF > 100$) as fixed effects. The resulting parameter estimates of the full model, containing all factors as fixed effects, showed that participants' RT increased if they had to sort the sequence, as compared to just remembering it ($\hat{\beta} = 0.26$, 95% HDI=[0.17, 0.36]). Furthermore, the RTs increased for longer sequences ($\hat{\beta} = 0.66$, 95% HDI=[0.57, 0.82]). Thus, our first hypothesis was confirmed.

In line with our second hypothesis, we found that the latent structure had an effect. Participants responded faster in the *query structure* condition ($\hat{\beta} = -0.26$, 95% HDI=[−0.41, −0.12]), and in the *sequence structure* condition ($\hat{\beta} = -0.17$, 95% HDI=[−0.25, −0.09]) when compared to the *no structure* condition. (see Fig. 2B and table 1 for a summary of the model estimates). Thus, our sec-

ond hypothesis was also confirmed.

Yet one concern might be that since we used a within-subject design to compare the behavior of the same participants on all tasks, it is possible that some participants simply improved over the blocks, resulting in spurious effects. Accordingly, to make sure that the observed effect of structure was not due to block order effects, we included the block number as an additional fixed effect in the model (the block number was already included as a random effect in the previous model). We found that with increasing block number the RTs were reduced ($\hat{\beta} = -0.09$, 95% HDI=[−0.12, −0.07]). However, the effect of the *query structure* only seemed to increase with the inclusion of the blocks ($\hat{\beta} = -0.33$, 95% HDI=[−0.44, −0.22]), while the effect for *sequence structure* remained approximately the same ($\hat{\beta} = -0.16$, 95% HDI=[−0.22, −0.11]). The effect of the sequence length also remained unchanged ($\hat{\beta} = 0.66$, 95% HDI=[0.60, 0.73]). This indicates, that participants learned to sort faster over the blocks, but that learning alone cannot explain our results.

Another potential concern lies in the possibility that parts of people's mental sort happened during recall. If this was the case, we would be neglecting part of the sorting process in our analysis. To investigate this, we analysed the trade-off between encoding RTs and recall RTs. If people continued the sort in the recall RT, then the trials in which this happened should have shorter encoding RTs, resulting in a negative correlation between the two. Instead we found an overall positive correlation between the two. Even when accounting for different sequence lengths or structures, this relationship remained positive for almost all scenarios (see Fig. A1B). Additionally, we also ran a full model with all RTs (encoding and recall), with the RT-type as an interaction effect, the results of which again supported the idea, that the sort did not happen in the recall RT (see appendix A for more details). We, therefore, concluded that participants did not deliberately push any sorting behavior into the recall part of our experiment.

In summary, we found that participants' encoding RT increased with the length of the sequence and benefited from latent structure. In the next section, we will further investigate how participants' encoding RTs increased with longer sequences.

Scaling analysis

Having shown that there was a measurable difference between the *sort task* and the *memory task* conditions, we investigated how sorting times scaled with increasing input size by analyzing the difference between these two conditions (see Fig. 3A). For the following analysis (unless otherwise stated), we used the difference between the two tasks (i.e. Sort RT - Memory RT), which we refer to as sorting time. Since the trials were made to match each other in the two task conditions in length and queried position, we only included the differences where both trial types met the exclusion criteria (i.e. the response was correct and the RT was below 10s in both the memory and the sort task), leaving us with 6193 differences. Because we later also calculate differences in the sorting times between the different sequence lengths, and this cannot be done on a trial by trial basis, we used the summarized data of each sequence length and structure per participant for all analysis in this section.

Sorting time scales linearly for the given sequence lengths. To investigate whether the increase in sorting times was linear, sub-linear, or super-linear, we combined two comparisons.

In the first comparison, we transformed the sequence length to represent different complexities (from constant to exponential scaling, see below for details). This allowed us to investigate which complexity best described participants' sorting times. For this purpose, we calculated maximally-structured mixed effects models on the sorting times. The models contained the structure condition and the sequence length (s) as both fixed and random effects over participants (because we used the differences, which covered both task conditions and were calculated from trials that came

Table 1
Fixed effects of the full model of the encoding RTs.

	Encoding RT	
Predictors	Estimate	HDI(95%)
Sequence Length	0.66	0.57, 0.82
Query Structure	−0.26	−0.41, −0.12
Sequence Structure	−0.17	−0.25, −0.09
Sort Task	0.26	0.17, 0.36
Intercept	0.60	0.38, 0.82
Observations	13,444	
$N_{subjects}$	73	
Marginal R^2 / Conditional R^2	0.357 / 0.529	

This Table summarized the model results of the full model of the encoding RTs. This is the same model that is also shown in Fig. 2B.

from different blocks, we could not include the blocks or the task condition as factors in this analysis). To cover the space of different complexities, we applied different functions f to s. As such, we had a constant model, a logarithmic model, a linear model, a polynomial model (2nd degree), and an exponential model. These functions were defined as follows: constant: $f_{const}(s)=1$; log: $f_{log}(s)=log_{10}(s)$; linear: $f_{lin}(s)=s$; polynomial(2): $f_{poli}(s)=s^2$; exponential: $f_{exp}(s)=e^s$. We would have also liked to test the complexity of $N \log N$. However, since in the space of 1-7 $N \log N$ behaves very similarly to linear functions (depending on the slope and the base of the logarithm), we decided to exclude this model (see Fig. E1, for the results of that comparison). We then did a model comparison by calculating the BFs of all pairwise model-combinations (see Fig. 3B for a depiction of all results). The only model that was better than all others was the linear model (Linear vs. Constant: $BF > 100$; Linear vs. Log: $BF > 100$; Linear vs. Polynomial(2): $BF > 100$; Linear vs. Exponential: $BF > 100$).

Hence, the first comparison supported the notion that participants' sorting times scaled linearly. In our second comparison, we looked at an approximation of the derivative of scaling times over the sequence length. To calculate this approximation, we took the differences between each participants' sorting time for n rectangles and $n+1$ rectangles for all consecutive elements of n (we excluded all participants that did not have valid trials for all seven sequence lengths). The rationale of this analysis is that the derivative of a linear function should be constant, and, therefore, regressing n onto this difference should not improve the model fit com-

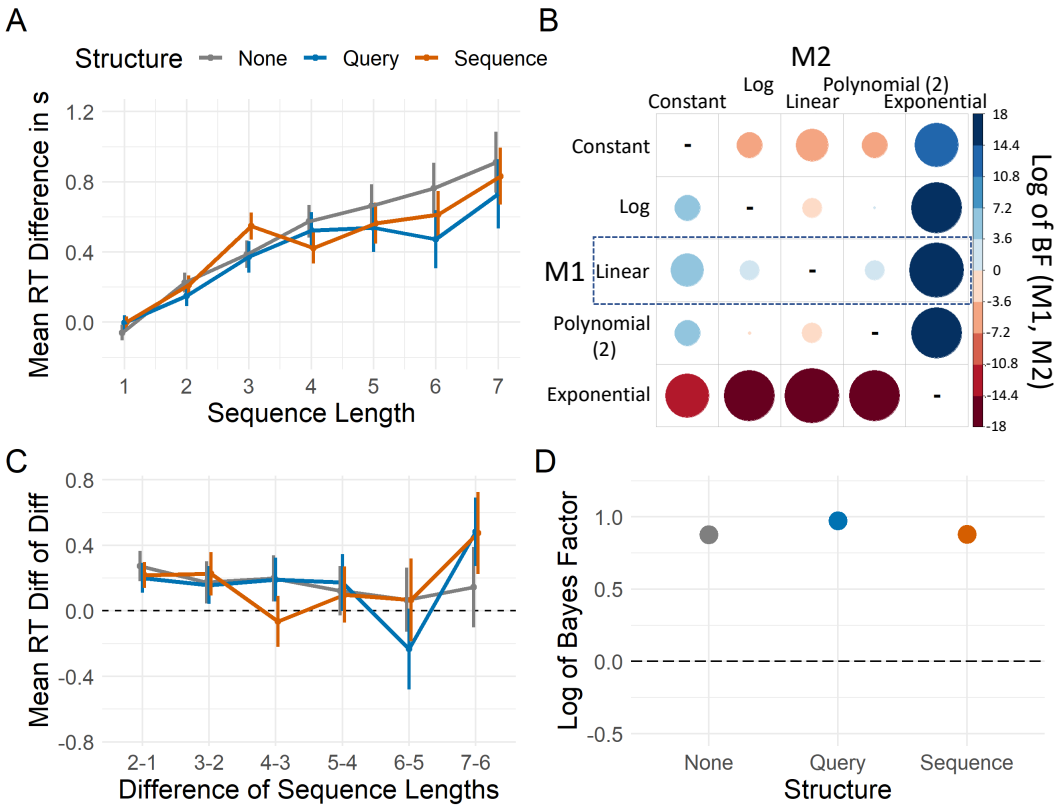

Figure 3. Evidence for Linear Scaling. **A**) The y-axis shows the differences of the *sort task* RTs and the *memory task* RTs. This differences represents the sorting time (without the memory component). **B**) Scaling analysis. We calculated maximally-structured mixed effects models on the RT difference depicted in **A**. The model contained the structure condition and the sequence length (s) as both fixed and random effects over participants. To compare various complexities in which the sequence length s could affect the RTs, we applied different functions f(s) to the sequence length (i.e. constant: f_{const}(s)=1; log: f_{log}(s)=log_{10}(s); linear: f_{lin}(s)=s; ploynomial(2): f_{poli}(s)=s^2; exponential: f_{exp}(s)=e^s). We depict the log of the BFs, meaning positive values (blue) give evidence for a model and negative values (red) give evidence against it. The size and the hue of the circle represents the size of the evidence. The rows represent the models for which the evidence is gathered, meaning that the winning model is the model with where the whole row has values above zero. **C**) Sorting time increase for each sequence length increase. The plot shows the mean of the difference of the values shown in **A** from each s to the next larger s +/- SE. This difference of differences is akin to a derivative: it should be 0 for constant scaling, constant for linear scaling and above constant for super-linear scaling. **D**) Evidence in favour of linear scaling. For each structure we calculated a constant and a linear model trying to predict the differences of the differences displayed in **C**. The BF here is log-transformed (as in **B**) and represent the evidence in favour of linear scaling.

pared to an intercept-only model. If including n as a predictor does, however, improve the model fit (i.e. the derivative is not constant), then this would be evidence that the sorting time scaled super-linearly. The derivative should be 0 if the scaling was constant. For the same reasons as above,

the only random effect we included in the model was *n*. Furthermore, to test our hypothesis that the scaling for the structure conditions should be better, we calculated a separate model for each structure. Our results showed that for all structures, we find evidence for linear scaling (i.e. the constant model performed better than the model containing *n*). However, for all structure conditions, the $\hat{\beta}$-estimate of *n* overlapped with 0, leaving open the possibility of sub-linear scaling (though the possibility of constant scaling has been excluded by our analysis above; *No structure* condition: $BF = 7.54$, $\hat{\beta} = -0.03$, 95% HDI=[−0.11, 0.05], *query structure* condition: $BF = 9.67$, $\hat{\beta} = 0.01$, 95% HDI=[−0.08, 0.09], and *sequence structure* condition, $BF = 7.43$, $\hat{\beta} = 0.03$, 95% HDI=[−0.06, 0.11]).

To summarize, we found evidence that mental sorting in our task likely scaled linearly or perhaps even sub-linearly.

The effects of structure

As we proposed in our second and third hypotheses, one reason why human cognition could scale to complex problems is because humans recognize and exploit structural regularities in the environment. Our behavioural results already showed that participants used the latent structures in our task to improve their RTs (hypothesis 2). In the next part we tested our third hypothesis, by investigating what exactly this improvement looked like and whether it aligned with our expectations regarding the used structures.

Structure helps, but is not used to its full extent. We first calculated a model in which we included an interaction effect of sequence length and structure. We found an interaction between *query structure* and sequence length, resulting in larger RT decreases for longer sequences. For the *sequence structure* there seemed to be a small effect in the same direction, but the results were less clear (see Table D1). To quantify the effect of structure further and to test our third hypotheses (namely, that participants would profit increasingly with increasing sequence length in the *query structure* condition and that the encoding RTs would be faster by a constant amount for the *sequence structure* condition), we calculated the differences between the no structure sort task and the two structure sort tasks (see Fig. 2C). If people really used the structure, we would expect there to be an increasing difference between the *no structure* condition and the *query structure* condition, since the longer the sequence, the more people should benefit from not having to sort it. For the *sequence structure* condition we expected the difference to increase for the first three rectangles and then stay constant, since there are only three connected rectangles and otherwise the sorting is the same as for the *no structure* condition. To test these hypotheses, we ran three models on the two differences between the conditions. The first model was a constant (intercept-only) model (representing the hypothesis that there was no or a constant difference), the second model had the sequence length as a predictor (representing our hypotheses about the benefit of the *query structure* condition) and the third model also had the sequence length as a predictor, but coded sequence lengths above 3 as 3 (representing the hypothesis about the *sequence structure* condition).

For the *query structure* condition, we found that including the sequence lengths improved the model, both compared to a constant model ($BF = 50.85$) as well as to a model with the re-coded sequence length ($BF > 100$). This indicates that people used the *query structure* with increasing benefits for longer sequences. For the *sequence structure* condition, however, the best model was less clear. Both the intercept only model and the re-coded sequence length model were better than the model with the normal sequence length ($BF = 2.33$ and $BF = 1.97$), and the constant model was better than the re-coded model ($BF = 1.2$), but the BFs were comparatively small. Nonetheless, these results suggest that people did not benefit as much from the *sequence structure* as we expected.

In summary, as we proposed in our third hypoth-

esis, *query structure* increasingly benefited participants RTs for longer sequence lengths. However, while we have shown in previous analyses that there was also a benefit for the *sequence structure* condition, this benefit was smaller and did not take the form we expected.

Models of Structure learning

To investigate the mechanisms people used to learn latent structure, we evaluated two potential models of participants' behavior. Since for this analysis we focused on capturing the mechanisms that people used to learn latent structure to inform their mental sorting, we chose one sorting algorithm (as a stand-in for any sorting algorithm that scales linearly) which matched the linear scaling we observed empirically. Specifically, both models were based on a bucket sort algorithm (Horsmalahti, 2012), which is not an exact comparison-based algorithm and, therefore, achieves better scaling in exchange for being prone to noisy errors. Our bucket sort algorithm takes knowledge about the range of possible sizes of the rectangles into account to immediately sort each rectangle into the correct bucket (see appendix B for more details).

To benefit from latent structure, an agent needs to propose and evaluate hypotheses about the structure of the task. We assume that hypotheses about structure can contain information about three things: 1) which rectangles might be connected, 2) how long to sort, and 3) which sort direction is more beneficial (i.e. one example hypothesis would be a connection between the "red" and "blue" rectangles, a sort length of three and a sort that start with the smallest rectangle). The evaluation of a proposed hypothesis can be performed based on whether or not the resulting sorting process was correct and how long it took. We looked at two models (see Fig. 4A) which varied in their search method, determining which hypotheses were currently evaluated for their usefulness. In other words, the search method defined how an algorithm proposed hypotheses about the structure in the task. We did not strictly model the complexity of the process of learning the structure, since that can be done offline (not during a trial).

Hypothesis mutator. We first considered a model with undirected search, using random mutations to traverse the space of possible hypotheses. This model used an evolutionary search model, which evaluates a limited set of hypotheses about the structure, exchanging bad hypotheses (i.e. hypotheses that either resulted in wrong responses or that were correct, but slow) with mutated variants of better performing hypotheses. Meaning this model has two hyper-parameters. 1. the number of evaluated hypotheses at any given time and 2. the number of hypotheses which get replaced with mutants. Because the number of evaluated hypotheses was fixed, the computational costs of this model remained constant with the amount of possible hypotheses, but plausible hypotheses were harder to locate.

Hypothesis generator. In contrast, we also developed a model using directed search, based on regularities in the sorted sequences and the queried positions to generate a plausible hypothesis. The generator only considered one hypothesis at a time, which was changed based on representations of transitions between colors T, the maximum queried position $\mathbf{b}$, and the best sort direction $\mathbf{d}$.

The transition matrix T represented transitions between colors in the sorted sequence (i.e. the probability that "red" follows "blue") and was updated after each trial with the observed transitions in the sorted sequence. If the probabilities of certain transitions exceeded 0.8, the generator grouped the concerned colors together in future trials, eliminating the need to sort the rectangle with the second (or third) color. The threshold vector $\mathbf{b}$ encoded the maximum position of the queried position, such that if the second position kept being queried, this gradually formed the hypothesis that only two rectangles needed to be sorted. Lastly, the sort direction vector $\mathbf{d}$ encoded the sort direction based on the relative position that was queried. If the second position was queried in a sequence of

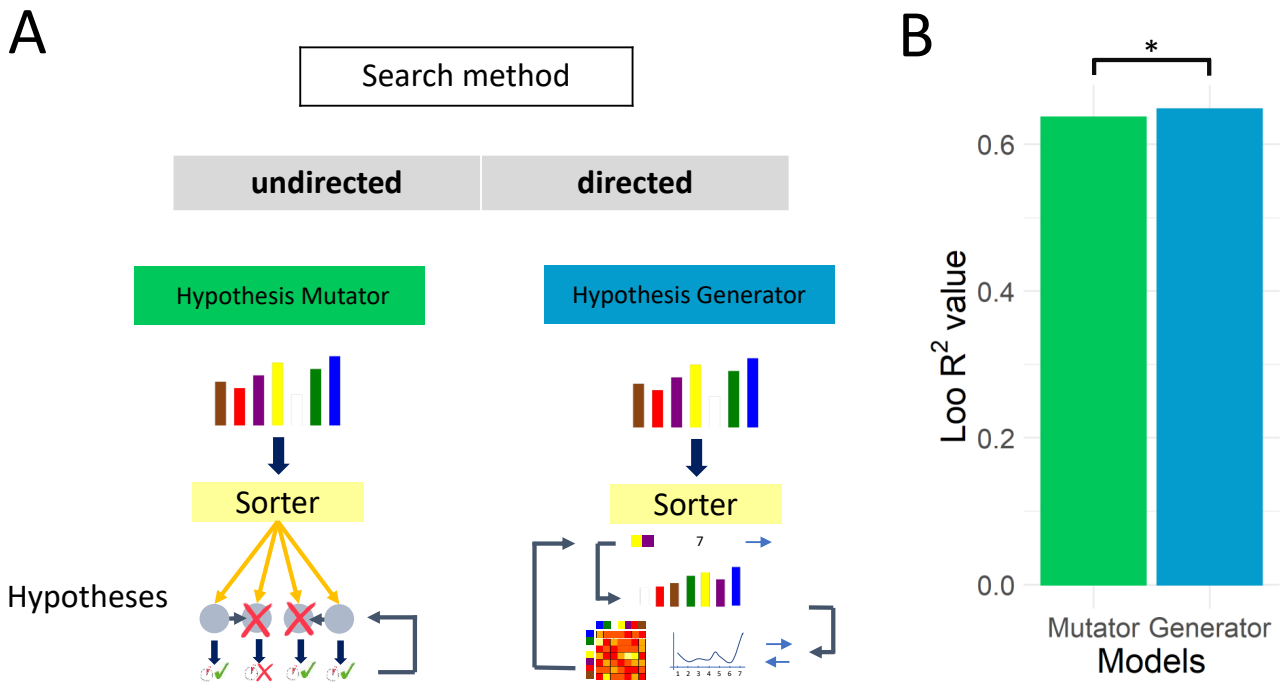

Figure 4. Sorting model. **A**) The two types of Models. The division illustrate the dimension on which the models differ. The illustration below depicts a schematic of the two different models. **B**) Model Comparison. This plot shows the loo R^2 values for an analysis where we tried to model the true RTs with two models that contained the model output of each of our models as well as the structure as both random and fixed effects. The predictions of the hypothesis generator explain most of the variance within the RT data. The * indicate $BF > 100$ in favor of the model with the higher loo R^2 value.

length 4, than this increased the probability of the model sorting the next sequence from the smallest to the tallest rectangle vs. from the tallest to the smallest rectangle. For details of the implementation of the models, see appendix B.

Models comparison. The two models we compared correspond to two different assumptions of how people search through hypotheses in order to use structure: random (hypothesis mutator) vs. directed (hypothesis generator). To ensure a fair model comparison, we used a grid-search over model parameters to determine the parameters that resulted in the highest log likelihood estimate for each participant.

We estimated the two models on trials from the *sort task* that each of the 73 participants observed. This means that the trials from each block (three blocks per participant, one for each structure) were fed into the models, which then generated sorting times for each trial of the separate blocks (for each block, the model started as a naive sorting algorithm, i.e. with the initial settings). We ran a Bayesian regression model on participants' encoding RTs of the *sort* task using these model times (as well as the structure) as fixed and random effects. We calculated the loo R^2 values to compare the three models. The hypothesis generator explained the most variance in the data (hypothesis generator: loo R^2=0.65, hypothesis mutator: loo R^2=0.64; see Fig. 4B and Fig. 5F). Since the difference between the loo R^2 values was small, we wanted to make sure, that it was nonetheless meaningful. Thus, we also compared the models directly. This comparison showed that the hypoth-

esis generator described participants' behavior better than the hypothesis mutator ($BF > 100$). Furthermore we also assessed if the hypothesis generator's behavior matched participants' behavioral patterns in the next section.

Models output. To make sure that the hypothesis generator is a valid description of human behavior, we qualitatively compared the model times we had generated (see above) to the data collected in our task. We found that the hypothesis generator generated human-like scaling patterns (see Fig. 5A and B) and that it also profited from the underlying structure. The hypothesis generator also replicated the empirical finding that participants benefited more from the *query structure* than from the *sequence structure*.

The hypothesis mutator, on the other hand, was unable to reproduce the full pattern of human RTs in our task. Specifically, the hypothesis mutator often learned faulty sequence structures, and was unable to replicate participants' use of latent structure, particularly the use of the *sequence structure* (see Fig. B1), making the hypothesis generator a better generative model of participants' behavior.

Interestingly, the hypothesis generator even improved its processing time in the *no structure condition*. It did this because it learned that sorting smaller parts of a sequence could still result in high accuracy given that items further down in the sorted sequence were queried only infrequently (this can be seen by the low thresholds for all conditions, see Fig. 5D), and the improved processing times for all conditions, see Fig. 5C). It is, therefore, possible that human subjects applied a similar strategy, decreasing their sorting time even in the *no structure* condition. This is a noteworthy finding because the model highlights a structural property of our task that could have been used by humans as a strategy to reduce sorting times.

Taken together, these results indicate that participants likely used a directed search method that took the observed transitions into account to generate hypotheses about latent structures in our task.

Discussion

People are robust to the varying complexities they encounter in everyday life. Yet cognitive models do not always scale as well with increasing complexity. To help bridge this gap, we proposed that studying the scaling of mental computations can be used to help identify plausible models of human cognition. We used RTs to assess the scaling of one such mental computation: mental sorting. We found that participants' sorting times scaled linearly with the number of rectangles they needed to sort. Additionally, participants recognized and actively exploited latent structure to improve their sorting times. To understand how this structure could be learned, we used computational modelling to compare two models that used undirected or directed search methods to learn hypotheses about the latent structure. We found that the participants' behavior was more in line with a model that applied directed search to generate and test hypotheses about underlying structures and that this model was able to replicate our observed behavioral patterns. Taken together, these results show that people deal well with increasing complexities (at least at the scale presented in our experiment) and open up new avenues to study how mental computations scale more generally.

As our study was a first attempt at understanding how mental computations scale with increasing input size, there are some limitations which still need to be addressed. One limitation of the current study is the length of the considered sequences. Since we required participants to remember the sequences, we were limited to a length which could still be maintained in working memory. And while the length of at most seven rectangles in a sequences is enough to provide us with the ability to make inferences about the super- or sub-linearity of the scaling, it does not enable us to make very fine-grained statements about the exact nature of the scaling. For instance, a complexity of $N \log N$ behaves favourably for short sequences and is therefore hard to distinguish from linear scaling. Furthermore, we cannot be certain how human scal-

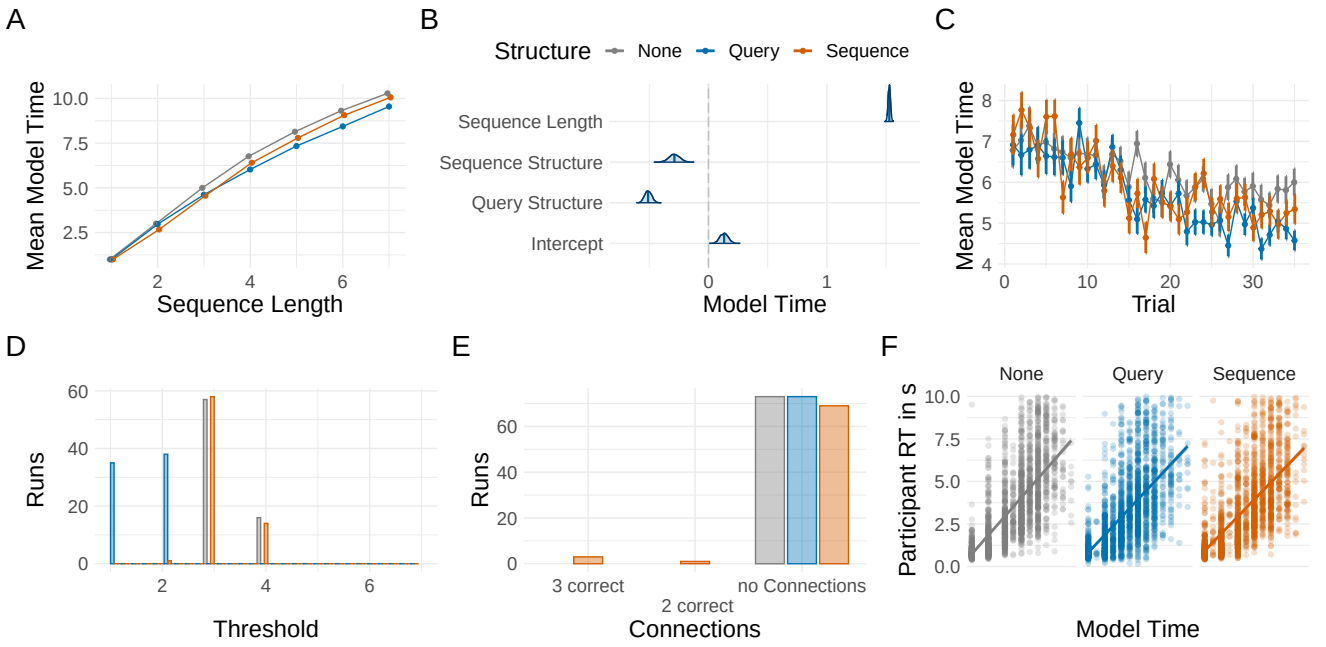

Figure 5. Output of hypothesis generator. **A**) Model time scaling. This plot shows the mean model time for each sequence length over all trials. **B**) Behavioural pattern. With this model we investigated whether the times of the hypothesis generator have a similar pattern to the human RT data depicted in (see Fig. 2B). The model thus had the same structure. **C**) Learning progress. This plot shows the improvement of the model times over the 35 trials of a block. **D**) Learned thresholds at the last trial. This plot shows how many rectangles the hypothesis generator was willing to sort for the last trial of each block. **E**) Learned connections at the last trial. This plot shows whether the hypothesis generator was able to learn the correct connections for each of the blocks. As can be seen, no wrong connections were learned. **F**) relation between model and behavioural data output. Since we ran the model on the same trials that the participants saw, we can plot the relationships of the model and the real RT data for each trial. This is depicted here.

ing behaviour changes if the sequence length is increased even further. It could be plausible that the sorting algorithm used by humans changes depending on the length of the sequences and that thus the scaling also differs for longer sequences. In fact most default sorters of programming languages also combine different sorting algorithms depending on the length of the to be sorted lists (e.g Timsort, which is the python default sorter, combines insertion sort for small lists with merge sort). Moreover, observing the complexity for sorting 1 to 7 rectangles also could lead to conclusions of linear scaling if the first few elements of the complexity curve look linear but indeed just mark the beginning of an exponential or logarithmic curve. Future studies could look at scaling times for more complex tasks to test the limits of this approach.

One important observation in our study is that participants make mistakes and the number of the mistakes increases with the sequence length. This can broadly be the result of two explanations with very different implications. First, the decreasing accuracy could be just a reflection of an increased difficulty to encode or recall the correct order of the rectangles for longer sequence. This would mean that the participants still tried to always fully sort the sequence. Therefore, this should not affect the

presented analysis of our results, where we indeed assumed a full sort. This explanation is supported by the fact that errors increased in the *memory task* as well. A second possibility is that participants strategically reduced the number of rectangles they sorted in longer sequences. This would result in mistakes. However, since longer sequences are rare and even an incomplete sort had good chances of resulting in correct responses, the number of the mistakes would still be limited. As such, participants might have been willing to allow for these mistakes to reduce the overall workload. Our results do not allow us to conclusively differentiate between these two possibilities. Therefore, we cannot exclude the possibility that the favourable scaling of participants' sorting time is a result of incomplete sorts for longer sequences. However, since the effect of structure on RT and accuracy correlated positively over participants, there is at least some evidence that there is no trade-off between accuracy and sorting time (see appendix C). This makes it unlikely that some participants explicitly accepted a lower accuracy to reduce their sorting time.

In relation, it needs to be mentioned that in our current design we only rewarded accuracy. This could be problematic in two ways. First, it could mean that participants abandoned strategies which are fast, but which have some (acceptable) degree of error. Secondly, it is also possible that by rewarding accuracy we motivated participants to be extra cautious, and thus the RTs might not only reflect the sorting time, but also an added time factor due to cautiousness (which would, however, only be problematic if this extra time also scaled with the input size).

Another point to consider is the relationship between memory and sorting. While we introduced the *memory task* to be able to abstract away everything that was not sorting from the analyzed response times, with the present study, we are unable to confirm that this is a valid analysis. It is possible that memory and sorting are not additive processes, but rather that they interact. Memory and sorting could, for example, be sharing some common resource and therefore interfere with each other especially for longer sequences. This would also result in longer response times. As such we can not be sure that the increase in RT actually represented the complexity of mental sorting. The fact that we do observe the pattern in increase of RTs with increasing sequence length, while controlling for increases in RTs from the memory task, does, however, support the notion that the RTs are related to the length of the actual mental sorting process.

One question that our study has not yet addressed concerns the exact algorithms used by participants to accomplish linear scaling. Given the fact that all exact comparison-based sorting algorithms scale super-linearly, it is remarkable that participants' mental sorting time scaled linearly. Our study, therefore, provides evidence that humans are not using exact, comparison based sorting algorithms. However, this result still leaves various possible algorithms to consider. We used a bucket sort algorithm in our model, which explained participants' behavior well. By avoiding the pairwise-comparison, this algorithm can be error-prone, just like we observed in participants' behavior. However, bucket sort is not the only possible algorithm that scales linearly. Other sorting algorithms with the same complexity could be just as likely given our current results. For example, other similar algorithms with subtle differences like radix sort, or counting sort (Horsmalahti, 2012) could also work. Another promising option would be a parallel sorting mechanism which functions like a criterion-bar that is moved either up or down all rectangles at the same time. Rectangles which exceed (in case of upward movement) or are below (in case of downward movement) the current position of the bar are then sequentially moved into the next available position of the sorted sequence. However, a different study design, which probes the idiosyncrasies of different sorting algorithms, would be required to make a clearer statement about which of these algorithms

is most likely. The aim of our current study was not to identify the exact algorithms with which humans solved our task, but rather to use the scaling complexity as a criterion, with which we can evaluate the plausibility of a wider range of algorithms or models in future investigations.

Our modeling results support the notion that participants used a directed search method that was informed by the observed transitions to generate hypotheses about latent structures. The incremental way in which the hypothesis about the current structure is generated reminiscent of previous research. For instance, Bramley, Dayan, Griffiths, and Lagnado (2017) proposed that structure can be learned by maintaining a global hypothesis, which is updated via local changes, illustrating an unwillingness to abandon the current hypothesis about the structure entirely. The hypothesis generator functions similarly, by taking the properties of the current trial into account to slightly adjust the believe about the underlying structure. Furthermore, the way in which the model learned the *sequence structure*, was inspired by existing sequence learning models (Éltető, Nemeth, Janacsek, & Dayan, 2022), though due to the deterministic nature of our structures, our version is relatively simple in comparison. In less deterministic environments the model would likely need to be adjusted accordingly. In summary, our model suggests, that participants behaviour is well described by a model, which constructs one global hypothesis, which is sequentially updated based on the encountered sequences. This generation process does not explicitly reward speed (as opposed to the hypothesis mutator), but nonetheless results in faster processing times for the structure conditions. However, in this study we only compared two models as broad representatives of a directed or undirected search across all possible structures. Further studies are necessary to delineate more precise mechanisms by which latent structure can be learned in tasks like this.

Finally, we believe that other psychological domains could also benefit from gaining further insights into the scaling of the computations of the concerned mental processes. And while we have currently only applied our approach to a simple mental sorting task, we would like to study other domains, such as category learning or retrieval from long-term memory, using a similar approach in the near future. To further arbitrate between different process-level models of mental computations, one could also combine our current approach with additional method to gain insights about what people do and attend to. Two such methods could be eye-tracking to assess where people look at while solving a task (J. R. Anderson & Douglass, 2001) or MEG to decode their programming traces when applying a particular algorithm (Eldar, Lièvre, Dayan, & Dolan, 2020).

Related Work

Our work builds on a rich tradition of linking mental computations and reaction times. Apart from the aforementioned studies on seriation (Young & Piaget, 1976) and mental rotation (Shepard & Metzler, 1971), reaction times have also been frequently used to measure how long people ponder before making a decision (Ratcliff, 1978; Ratcliff & McKoon, 2008). In this line of research, the theoretical shape of reaction times (Tejo, Araya, Niklitschek-Soto, & Marmolejo-Ramos, 2019) as well as how they can be used to compare cognitive and neuroscientific models has been discussed (Steinkamp, Fink, Vossel, & Weidner, 2022).

There also exist other studies on human sorting behavior. Lieder et al. (2014) studied how people choose between different sorting algorithms in a manual sorting task, showing that participants can be trained to either perform cocktail sort or merge sort-like behaviors after training them on such algorithms. Thompson, van Opheusden, Sumers, and Griffiths (2022) studied how participants sorted sequences of unknown numbers and how the resulting algorithms were shaped by cultural evolution, showing that several known sorting algorithms were discovered during cultural trans-

mission chains. Sorting has also been studied using the Wisconsin Card Sorting Task (Grant & Berg, 1993) in which participants need to sort cards according to one of three criteria: color, shape, or number of the designs on the face of the cards, while the experimenter changes the used criterion after the participant has made 10 consecutive correct classifications. This task has not only been used to study patients with brain damage (S. W. Anderson, Damasio, Jones, & Tranel, 1991), but also been analyzed using computational models of symbolic sorting algorithms (Dehaene & Changeux, 1991).

We are also not the first to show that participants benefit from repeatedly encountering structure in their environment. As studied extensively in the literature on practice effects, participants tend to reuse the solutions to previously performed computations to speed up their responses when the same problems are encountered again (Logan, 1988). However, directly reusing past solutions does not fully utilize the structure in the space of queries. While two problems or queries might not be exactly the same, they might have partial similarity that can be leveraged by more flexible reuse (Dasgupta & Gershman, 2021). This more flexible remembering and reusing of partial solutions– made possible by recognizing the structure in the space of queries is referred to as amortization of computation. Past work has shown that it is prevalent in human planning (Huys et al., 2015; Mattar & Daw, 2018), and it has recently also been studied in human probabilistic inference (Dasgupta, Schulz, Tenenbaum, & Gershman, 2020). Additionally, how people learn that certain steps in a computation can be skipped, as was the case in our sequence structure condition, has also been studied before, particular in mental algebra. For example, in a series of experiments conducted by Blessing and Anderson (1996), participants had to perform mental algebra to solve problems in which they could skip steps and still arrive at the correct solution. Their results showed that participants first skipped steps mentally but later started to use fully new transformations, thereby covertly skipping steps.

Conclusion

In summary, we have applied an approach towards testing the plausibility of psychological models based on the scaling of participants' response times to take a precise look at mental sorting. We found that mental sorting scales surprisingly well and that latent structure, is used to improve the time complexity for mental sorting. We believe that this approach will provide a widely-applicable and fruitful assay for future investigations.

Author Contributions

All authors developed the study concept and contributed to the study design. SH conducted the experiment and SH and ES performed the data analysis and interpretation. SH and ES drafted the manuscript, and CMW and ID provided critical revisions. All authors approved the final version of the manuscript for submission.

Acknowledgements

We thank Shuchen Wu and Noémi Éltető for feedback on earlier versions of our experiments and analyses.

Declaration of Conflicting Interests

The authors declared that they had no conflicts of interest with respect to their authorship or the publication of this article.

Funding

This work was supported by the Max Planck Society and a Jacobs Research Fellowship to ES. SH is supported by the Max Planck School of Cognition. CMW is supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A and funded by

the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2064/1 – 390727645.

Open Practices

All code and data have been made publicly available and can be accessed at `https://github.com/susanneharidi/mental-sorting`

References

Anderson, J. R., & Douglass, S. (2001). Tower of hanoi: Evidence for the cost of goal retrieval. *Journal of experimental psychology: learning, memory, and cognition*, *27*(6), 1331.

Anderson, S. W., Damasio, H., Jones, R. D., & Tranel, D. (1991). Wisconsin card sorting test performance as a measure of frontal lobe damage. *Journal of clinical and experimental neuropsychology*, *13*(6), 909–922.

Ashcraft, M. H., & Battaglia, J. (1978). Cognitive arithmetic: Evidence for retrieval and decision processes in mental addition. *Journal of Experimental Psychology: Human Learning and Memory*, *4*(5), 527.

Barr, D. J., Levy, R., Scheepers, C., & Tily, H. J. (2013). Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of memory and language*, *68*(3), 255–278.

Bartsch, L. M., & Oberauer, K. (2021). The contribution of episodic long-term memory to working memory for bindings. *PsyArXiv*.

Berg, E. A. (1948). A simple objective technique for measuring flexibility in thinking. *The Journal of general psychology*, *39*(1), 15–22.

Blessing, S. B., & Anderson, J. R. (1996). How people learn to skip steps. *Journal of experimental psychology: learning, memory, and cognition*, *22*(3), 576.

Bossaerts, P., & Murawski, C. (2017). Computational complexity and human decision-making. *Trends in Cognitive Sciences*, *21*, 917–929.

Bramley, N. R., Dayan, P., Griffiths, T. L., & Lagnado, D. A. (2017). Formalizing neurath's ship: Approximate algorithms for online causal learning. *Psychological review*, *124*(3), 301.

Brunswik, E. (1952). The conceptual framework of psychology. *Psychological Bulletin*, *49*(6), 654–656.

Bürkner, P.-C. (2018). *Advanced bayesian multilevel modeling with the r package brms. r j. 10, 395–411. doi: 10.32614* (Tech. Rep.). : RJ-2018-017.

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. doi: 10.18637/jss.v080.i01

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). "8". In *Introduction to algorithms, 3rd-edition* (p. 167). MIT Press and McGraw-Hill.

Crosby, S. A., & Wallach, D. S. (2003). Denial of service via algorithmic complexity attacks. In *Usenix security symposium* (pp. 29–44).

Dasgupta, I., & Gershman, S. J. (2021). Memory as a computational resource. *Trends in Cognitive Sciences*. doi: https://doi.org/10.1016/j.tics.2020.12.008

Dasgupta, I., Schulz, E., Tenenbaum, J. B., & Gershman, S. J. (2020). A theory of learning to infer. *Psychological Review*, *127*(3), 412.

Dehaene, S., & Changeux, J.-P. (1991). The wisconsin card sorting test: Theoretical analysis and modeling in a neuronal network. *Cerebral cortex*, *1*(1), 62–79.

De Leeuw, J. R. (2015). jspsych: A javascript library for creating behavioral experiments in a web browser. *Behavior research methods*, *47*(1), 1–12.

Eldar, E., Lièvre, G., Dayan, P., & Dolan, R. J. (2020). The roles of online and offline replay in planning. *Elife*, *9*, e56911.

Éltető, N., Nemeth, D., Janacsek, K., & Dayan, P. (2022). Tracking human skill learning with a hierarchical bayesian sequence model. *bioRxiv*.

Gershman, S. J., Horvitz, E. J., & Tenenbaum, J. B. (2015). Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, *349*(6245), 273–278.

Gigerenzer, G., & Brighton, H. (2009). Homo heuristicus: Why biased minds make better inferences. *Topics in cognitive science*, *1*(1), 107–143.

Gigerenzer, G., & Selten, R. (2002). *Bounded rationality: The adaptive toolbox*. MIT press.

Grant, D. A., & Berg, E. A. (1993). Wisconsin card sorting test. *Journal of Experimental Psychology*.

Griffiths, T. L., Lieder, F., & Goodman, N. D. (2015). Rational use of cognitive resources: Levels of analysis between the computational and the algorithmic. *Topics in cognitive science*, *7*(2), 217–229.

Gronau, Q. F., Singmann, H., & Wagenmakers, E.-J. (2017). bridgesampling: An r package for estimating normalizing constants. *arXiv preprint arXiv:1710.08162*.

Hamrick, J., & Griffiths, T. (2014). What to simulate? inferring the right direction for mental rotation. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 36).

Harlow, H. F. (1949). The formation of learning sets. *Psychological review*, *56*(1), 51.

Horsmalahti, P. (2012). Comparison of bucket sort and radix sort. *arXiv preprint arXiv:1206.3511*.

Huys, Q. J., Lally, N., Faulkner, P., Eshel, N., Seifritz, E., Gershman, S. J., ... Roiser, J. P. (2015). Interplay of approximate planning strategies. *Proceedings of the National Academy of Sciences*, *112*(10), 3098–3103.

Inhelder, B., & Piaget, J. (1958). *The growth of logical thinking from childhood to adolescence: An essay on the construction of formal operational structures* (Vol. 22). Psychology Press.

Kuroki, D. (2020). A new jspsych plugin for psychophysics, providing accurate display duration and stimulus onset asynchrony. *Behavior Research Methods*, 1–10.

Lieder, F., & Griffiths, T. L. (2020). Resource-rational analysis: understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, *43*.

Lieder, F., Plunkett, D., Hamrick, J. B., Russell, S. J., Hay, N., & Griffiths, T. (2014). Algorithm selection by rational metareasoning as a model of human strategy selection. *Advances in neural information processing systems*, *27*.

Lindsey, D. T., & Brown, A. M. (2014). The color lexicon of american english. *Journal of vision*, *14*(2), 17–17.

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

Logan, G. D., Ulrich, J. E., & Lindsey, D. R. (2016). Different (key) strokes for different folks: How standard and nonstandard typists balance fitts' law and hick's law. *Journal of Experimental Psychology: Human Perception and Performance*, *42*(12), 2084.

Lucas, C. G., Griffiths, T. L., Williams, J. J., & Kalish, M. L. (2015). A rational model of function learning. *Psychonomic Bulletin & Review*, *22*, 1193–1215.

Mattar, M. G., & Daw, N. D. (2018). Prioritized memory access explains planning and hippocampal replay. *Nature neuroscience*, *21*(11), 1609–1617.

McGonigle, B., & Chalmers, M. (2002). The growth of cognitive structure in monkeys and men. In *Animal cognition and sequential behavior* (pp. 269–314). Springer.

Papadimitriou, C. H. (2003). *Computational complexity*. John Wiley and Sons Ltd.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological review*, *85*(2), 59.

Ratcliff, R., & McKoon, G. (2008). The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation*, *20*(4), 873–922.

Schulz, E., Bhui, R., Love, B. C., Brier, B., Todd, M. T., & Gershman, S. J. (2019). Structured, uncertainty-driven exploration in real-world consumer choice. *Proceedings of the National Academy of Sciences*, *116*(28), 13903–13908.

Schulz, E., Tenenbaum, J. B., Duvenaud, D., Speekenbrink, M., & Gershman, S. J. (2017). Compositional inductive biases in function learning. *Cognitive psychology*, *99*, 44–79.

Shepard, R. N., & Metzler, J. (1971). Mental rotation of three-dimensional objects. *Science*, *171*(3972), 701–703.

Simon, H. A. (1990). Bounded rationality. In *Utility and probability* (pp. 15–18). Springer.

Steinkamp, S. R., Fink, G. R., Vossel, S., & Weidner, R. (2022). Simultaneous modeling of reaction times and brain dynamics in a spatial cueing task. *Human Brain Mapping*, *43*(6), 1850–1867.

Sternberg, S. (1969). Memory-scanning: Mental processes revealed by reaction-time experiments. *American scientist*, *57*(4), 421–457.

Tejo, M., Araya, H., Niklitschek-Soto, S., &

Marmolejo-Ramos, F. (2019). Theoretical models of reaction times arising from simple-choice tasks. *Cognitive neurodynamics*, *13*(4), 409–416.

Thompson, B., van Opheusden, B., Sumers, T., & Griffiths, T. (2022). Complex cognitive algorithms preserved by selective social learning in experimental populations. *Science*, *376*(6588), 95–98.

Van Rooij, I. (2008). The tractable cognition thesis. *Cognitive science*, *32*(6), 939–984.

Van Rooij, I., & Wareham, T. (2008). Parameterized complexity in cognitive modeling: Foundations, applications and opportunities. *The Computer Journal*, *51*(3), 385–404.

Wu, C. M., Schulz, E., Speekenbrink, M., Nelson, J. D., & Meder, B. (2018). Generalization guides human exploration in vast decision spaces. *Nature Human Behaviour*, *2*, 915—924. doi: 10.1038/s41562-018-0467-4

Young, R. M., & Piaget, J. (1976). *Seriation by children: An artificial intelligence analysis of a piagetian task*. Springer.

Appendix A

Results of recall RT.

To make sure that the sort did not happen in the recall RT, we ran a full model with the same predictors we used for the encoding RT model, but this time we also included the recall RT as an outcome variable and added the RT-type (encoding vs. recall) as an interaction effect. If the sort really was constrained to the encoding RT we would expect the *task* conditions to have no effect on the recall RT and for the *structure* conditions, only the *query structure* should be reducing the RT. The sequence length could still have an effect, since previous research shows that recall from working memory increases with the number of items that need to be remembered (Sternberg, 1969). The results of this analysis showed that recall RTs increased for longer sequences and decreased in the *query structure* condition, but not in the *sequence structure* condition. Surprisingly, the *sorting task* condition also increased the RTs, indicating that the sort influences the recall in some form (for results see: left column of table A1 and Fig. A1A). Accordingly, these results generally supported our conjecture, that the sort happens in the encoding RT. However, since the *sort task* had an effect, we also conducted a few follow up analysis, which are reported here:

1. If we found a trade-off between recall RT and encoding RT (i.e. that the recall RT was longer whenever the encoding RT was shorter), than this would support the idea, that some of the sort might be outsourced into the recall period. To investigate the existence of such a trade-off we looked at the recall and the encoding RT on a trial by trial basis. However, we found no evidence for a trade-off between encoding RTs and recall RTs (see Fig. A1B).

2. If some of the sort was happening in the recall RT, including the recall RTs as a fixed and random effects in the encoding RT model from the main analysis would potentially diminish the observed effects of sequence length and structure. Instead we found that the weights hardly changed. While the recall RT did have an effect, the participants' encoding RT still increased if they had to sort the sequence, as compared to just remembering it. The encoding RTs still increased for longer sequences. Participants still responded faster in the *query structure* condition, and in the *sequence structure* condition (see Fig. A1C and the middle column of table A1 for a summary of the model estimates)

3. Lastly, to make sure, that our reported results from the encoding RT are valid and that we are not missing anything which might be contained in the recall RT, we also model the sum of encoding RT and recall RT. None of the effects changed in any significant manner when compared to the results form the encoding RT analysis: The participants' encoding RT still increased if they had to sort the sequence, as compared to just remembering it. The encoding RTs still increased for longer sequences. Participants still responded faster in the *query structure* condition, and in the *sequence structure* condition (see Fig. A1D and table A1 for a summary of the model estimates).

Table A1
Model Results from the analysis of all RTs.

	Encoding RT		*Recall RT*	
Predictors	Estimate	HDI(95%)	Estimate	HDI(95%)
Sequence Length	0.60	0.52, 0.67	0.22	0.18, 0.26
Query Structure	−0.26	−0.39, −0.14	−0.09	−0.20, 0.02
Sequence Structure	−0.16	−0.22, −0.11	−0.02	−0.06, 0.02
Sort Task	0.23	0.15, 0.31	0.12	0.06, 0.18
Observations	26888			
$N_{subjects}$	73			
Marginal R^2 / Conditional R^2	0.321 / 0.486			

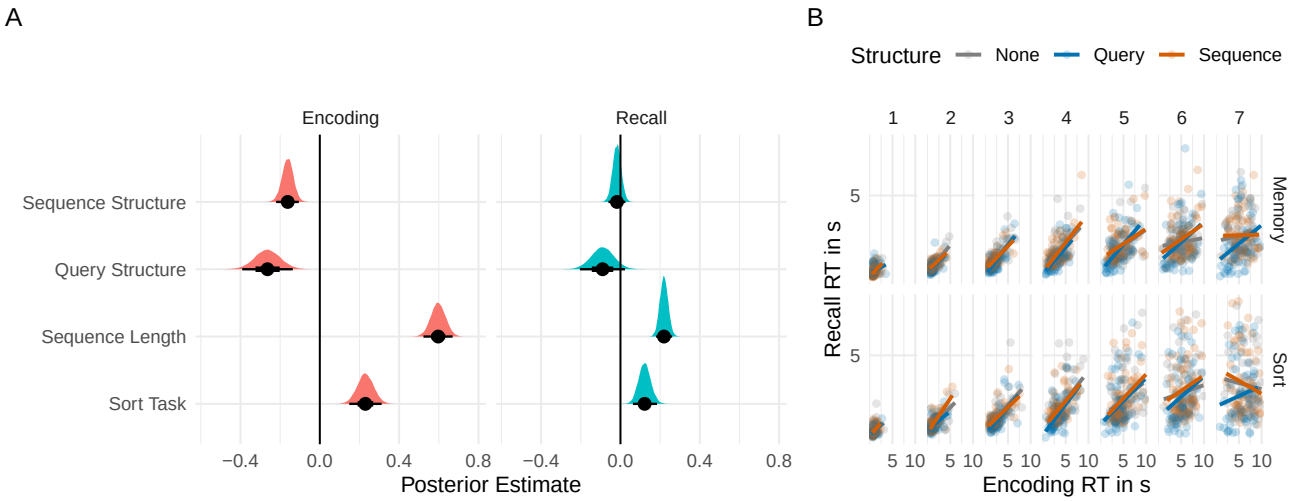

Figure A1. Recall RT. **A**) Model results of the model of all RT with RTtype as an interaction effect. This plot depicts the estimates of the effects of the full model all RTs. The model contained the displayed effects as both main and random effects and additionally also included the block number as a random effect. The main effects for the two RT-types are plotted separately here for easier interpretation. **B**) Trade-off between recall and encoding RT. The plot shows the relationship between encoding and recall RT. Each dot is the mean of one participants for all trials of the corresponding sequence length, structure and task. The columns represent the different sequence lengths and the rows the *memory* and the *sort task*.

Appendix B

The models

For comparison purposes,the two models were trained on the trials that the participants observed, however, the process of learning the latent structure was entirely independent from the participants actual behaviour on these trials and only the trial feature (such as sequence length or query position) mattered. Only after the model times were generated did we use the real RT data on these trials to do the hyper-parameter fitting as well as the final model comparison via fitting a Baysian regression which tried to predict the real RTs with the model time as a fixed and random effect over participants.

The sorter

To guarantee a linear scaling time for our sorter we implemented a bucket sort algorithm, where the sorting time is given by time=2*s (s: sequence length), if no Structure is learned. The sorter functions as follows: The sorter first sweeps through all rectangles once, while keeping track of the smallest (or largest) rectangle it has encountered so far. Since knowing the height of the smallest (or largest) rectangle determines the height of all other rectangles the sorter can now look at each bar and determine its exact position, meaning that the sorter only has to evaluate each rectangle twice to determine the position of all rectangles. We do not believe, that this is necessarily how humans sort, but it represents the linear scaling structure we encountered for humans. The direction of a hypothesis determines whether the sorter initially looks for the tallest or the smallest rectangle. The threshold determines after how many rectangles the sorting is stopped. All unsorted bars are then assigned in their current unsorted order to the positions which are not yet filled with sorted bars. And the connections of the sorter determine, which rectangles get grouped together. If such a grouping exists, the sorter only sorts the first rectangle in this grouping and adds the remaining ones only after the sort has finished. If the group does not actually represent the structure of the sequence, then this addition can occasionally cause subsequent, already sorted bars to be pushed into incorrect positions, potentially resulting in wrong responses.

Hypothesis mutator

The hypothesis mutator had two hyper-parameter: 1. the number of hypotheses (nH) which are maintained and 2. the number of hypotheses that get mutated (nM). For each participant we used a grid-search to determine the hyper-parameters resulting in the highest log-likelihood. After each trial all wrong hypotheses were eliminated. If no hypotheses were wrong, the slowest hypothesis was eliminated (the speed associated with a hypothesis was dependent on the speed on past trials, and speeds got inherited from the parent-hypotheses to their mutations). All "free" slots than got replaced by mutations of the fastest hypothesis. The mutation was either a mutation of the threshold or a mutation of the connection. Both of these had an equal probability of occurring. Threshold mutations moved the threshold up or down one element (e.g. from a threshold of 4 to a threshold of 5). For the connections several mutations were possible: 1. a color got removed, 2. a color got added (though the maximum number of colors was three), 3. a color changed randomly or 4. the connected colors got shuffled. The current model time was determined by the sort resulting from slowest hypotheses of all active hypotheses.

Hypothesis generator

The hypothesis generator had three representations that were adjusted after each observed trial and that allowed the model to learn the latent structure as well as to reduce its processing time. The size of

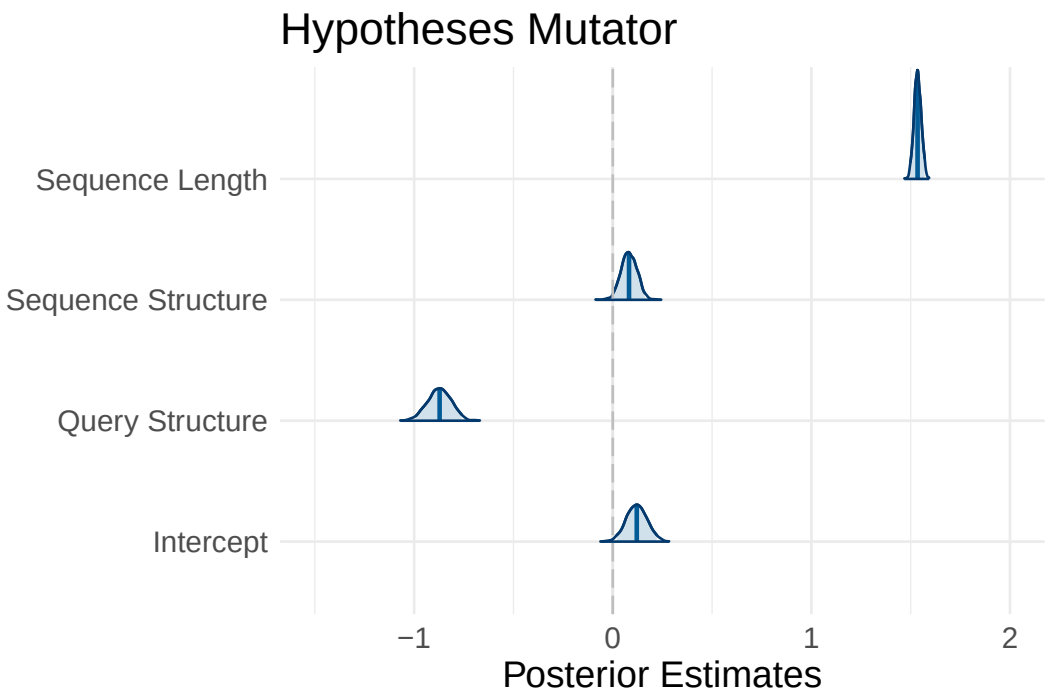

Figure B1. Behavioural pattern of the losing model, the hypothesis mutator. With this model we investigated whether the times (outcome variable) of the hypothesis mutator have a similar behavioural pattern to the human RT data depicted in Fig. 2B. The effect for the *sequence structure* did not match the human behaviour.

the adjustment depended on a learning-rate parameter α. For each participant we used a grid-search to determine the learning-rate resulting in the highest log-likelihood of the RT data given the model times. The three representations were:
1. An c x c transition-matrix T where c represents the number of colors we used (10 in our case). A particular cell represented the transition probability of the row-color to the column-color of that cell. Accordingly all cells in a row summed up to 1. After each trial the sorted sequence was used to update the transition-matrix according to the observed transitions. At the initialization of the model the matrix was agnostics, meaning that all values in the matrix are equal to 1/c. During the update the learning rate is added onto the observed cell corresponding to the observed transition, after which all values in the row are normalized, to maintain the property that the values in a row must add up to 1 (equation 2). This process is shown in equation 1 and 2:

$$y_{i,t+1} = y_{i,t} + \alpha \tag{1}$$

Here $y_{i,t}$ is the old value of the transition which is being updated and $y_{i,t+1}$ is the new value before normalization. In a second step (so that we have probabilities) we normalized the values of each element y_i in the row, so that the transition-probabilities for each row added up to one:

$$y_{i,t+1} = \frac{y_{i,t+1}}{\sum_{n=1}^{N} y_{n,t+1}} \tag{2}$$

$\sum_{n=1}^{N} y_{n,t+1}$ is the sum of all elements in that row. If one of the probabilities in the transition matrix exceeds 0.8, the corresponding transition was integrated into the sort (e.g. if the transition-probability between "red" and "blue" exceeds 0.8, the model will from then on sort all "blue" rectangles together with the "red" rectangles, without checking the actual size of the "blue" rectangle). The model can only have one such transition of at most three connected colors at any given time. If more than one transition meets this criterion at any given time, the model will select only one of these transitions to integrate in its sort. However, because the update of the transition-probabilities does not depend on a ground truth, but on the output of the current sort of the model, if a spurious transition exceeds the threshold by chance, the implementation above does not allow the model to correct for this mistake (as this spurious transition will always be part of the sorted sequence). Therefore the model updates differently for trials which resulted in wrong responses. If the sorter generated a wrong response, all values are updated by adding the learning-rate and then normalizing, increasing the overall uncertainty about transition-probabilities.

2. A 1 x k vector b, which determines the threshold, i.e. the maximum number of rectangles which are sorted, where k is the maximum sequence length (7 in our case). This threshold vector starts out with a probability of 1 for the threshold of 7 (i.e. sorting all rectangles in every sequence) and is updated after each trial according to which position was actually queried (the position is relative to the direction in which the model sorted in the given trial). This means that akin to equation (1) and (2), the learning-rate gets added onto the entry corresponding to the currently queried position. Afterwards the vector is normalized. If the model generated a wrong response the entry corresponding to the threshold of seven gets updated, increasing the overall threshold. The final threshold which the model used is the rounded mean of the k possible thresholds, weighted by the corresponding probabilities in the vector.

3. A 1 x 2 direction-vector d, governing the direction of the sort (i.e. if the sorter start from the smallest or the tallest rectangle). This vector was also updated according to the queried position (q). But instead of using the target position itself, we used the normalized distance to the mean length of the sequence (γ) to update the d. This was calculated as seen in equation 3:

$$\gamma = \frac{(s+1)/2 - q}{k} \tag{3}$$

with s being the sequence length. The update for the probabilities was then as follows:

$$P(d_1)_{t+1} = P(d_1)_t + \gamma\alpha \tag{4}$$

$$P(d_2)_{t+1} = P(d_2)_t - \gamma\alpha \tag{5}$$

d_1 and d_2 correspond to the 1st and 2nd entry of the direction-vector d, with d_1 being the probability of starting the sort from the smallest rectangle. Afterwards, d is again normalized akin to equation 1. If the model generated a wrong response we added the learning-rate onto both entries before normalizing to increase the general uncertainty about the direction. The model always learned to sort in the right direction for the *query structure* condition, which was the only condition, for which the direction of the sort mattered.

Appendix C

No time-accuracy trade-offs across subjects.

As mentioned in the discussion, one important question is whether participants were strategically trading of between accuracy and processing time. This could be achieved by reducing the number of bars which were sorted in each sequence. The consequence of this should be that the RTs are reduced, while the number of mistakes goes up (i.e. the accuracy goes down). However, since the accuracy is a binary measure, our study design did not allow us to look at accuracy-time trade-offs on a trial-by-trial basis. We did however, investigate, whether some participants had lower RTs and in exchange also had a lower accuracy in comparison to other participants, who were slower and more accurate. To do this we calculated two Bayesian regression models, one for the accuracy and one for the encoding RTs (we only used the trials from the *sort task* for this analysis and applied the same RT cutoff of 10s we used for our other analyses). Both models contained the sequence length as a fixed effect and random effect over participants. This allowed us to extract the individual effects that the sequence length had on both accuracy and RTs. If some participants sacrifice accuracy for shorter RT, we would expect to see a positive correlation between those two estimates. Instead, we found a small negative correlation ($r = -0.045$, Pearson's product-moment correlation), which was not significant ($p = 0.7, CI = [-0.27; 0.19]$). This indicates that no trade-off between accuracy and RTs is happening across participants.

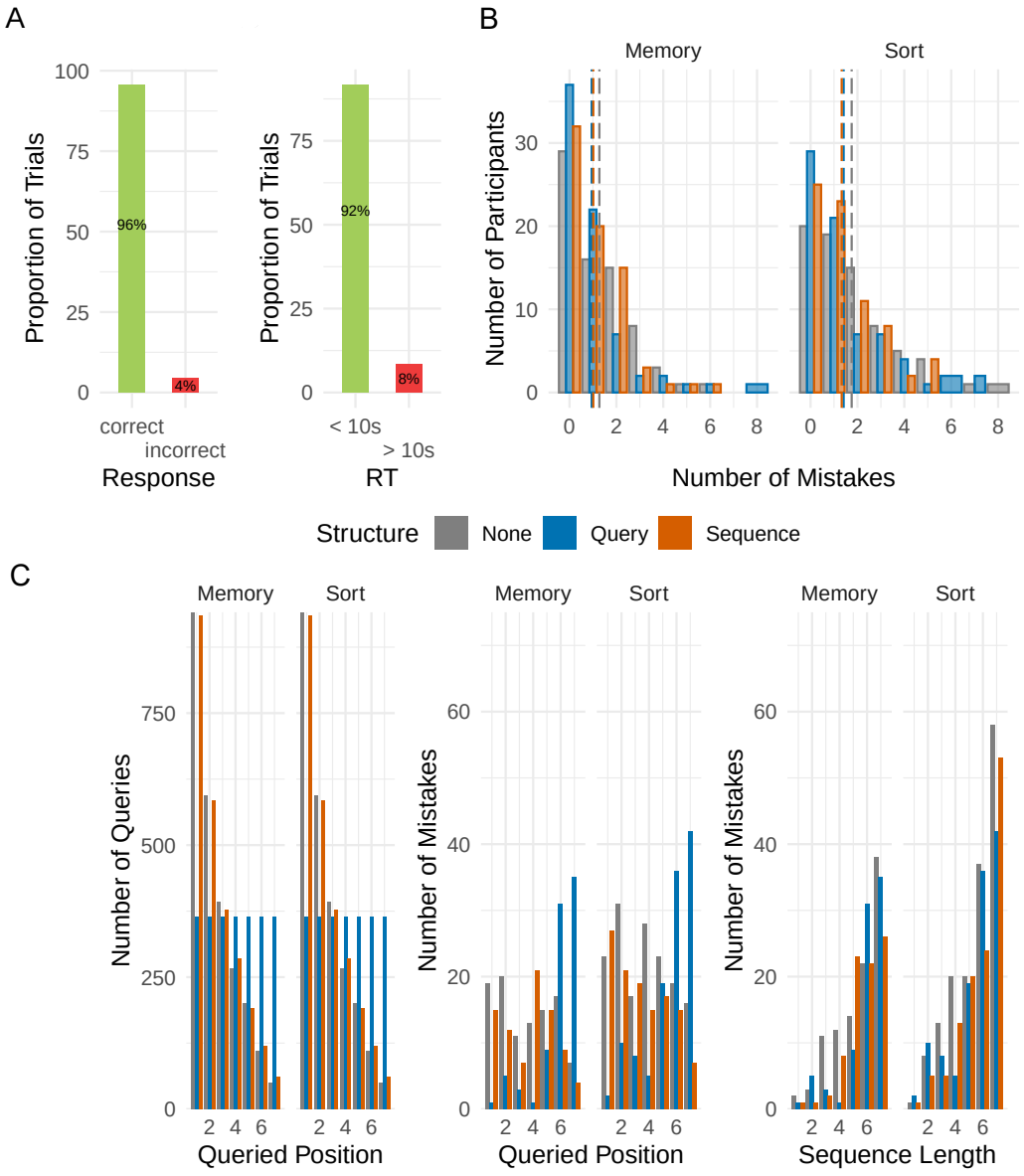

Figure C1. Accuracy. **A**) Excluded trials. The left plot shows the percentage of incorrect trials. The right plot shows the percentage of correct trials that exceeded the RT cutoff of 10 seconds. **B**) Error Distribution over Participants. The Histogram shows the distribution of the total number of Mistakes each participant made in the different Structure conditions over the number of rectangles. **C**) Properties of the errors. The first histogram, shows which positions get queried. this is a property of the task design and not of the participants behaviour. The reason the distribution is so skewed for all conditions (except the *query structure* condition, is because the queries were random and later positions can only be queried if the sequence is long enough. The second histogram shows the number of Mistakes, based on the position that got queried. The reason the *query structure* condition has more mistakes for the later positions is due to the fact that these positions also got queried more. The last plot shows the number of mistakes, based on the length of the sequence.

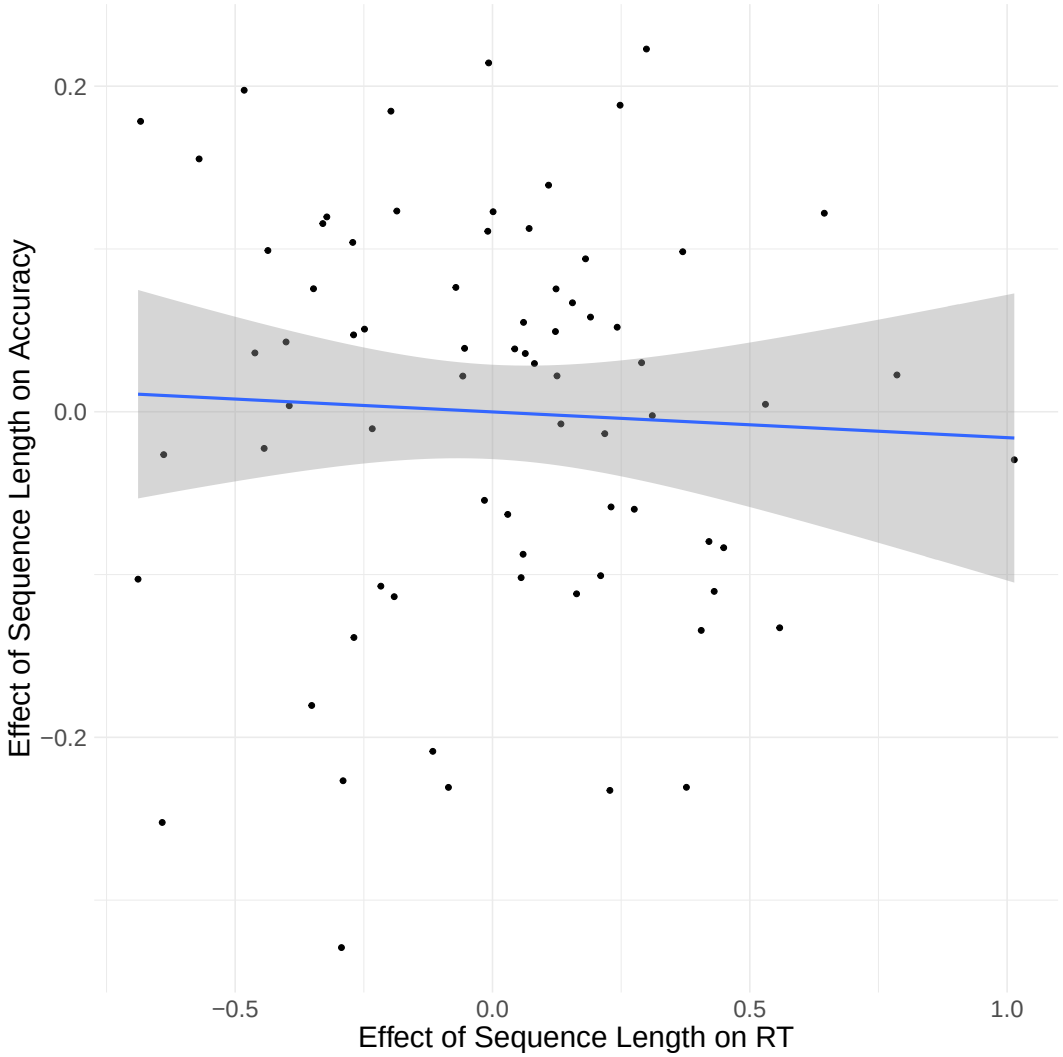

Figure C2. Accuracy-Time trade-off. The figures shows the relationship between the individual estimates of the effects of sequence length on accuracy and encoding RTs. Each data point is one participant.

Appendix D
Interaction Effects.

Table D1
Fixed Effects of the full model of the Encoding RTs.

	Encoding RT	
Predictors	Estimate	HDI(95%)
Sequence Length	0.66	0.56, −0.75
Query Structure	0.31	0.16, 0.45
Sequence Structure	−0.06	−0.16, 0.04
Sort Task	−0.19	−0.29, −0.09
Query Structure * Sequence Length	−0.16	−0.24, −0.08
Sequence Structure * Sequence Length	−0.04	−0.08, −0.00
Sort Task * Sequence Length	0.14	0.09, 0.18
Intercept	0.63	0.42, 0.83
Observations	13,444	
$N_{subjects}$	73	
Marginal R^2 / Conditional R^2	0.3622 / 0.546	

Appendix E
$N \log N$ scaling.

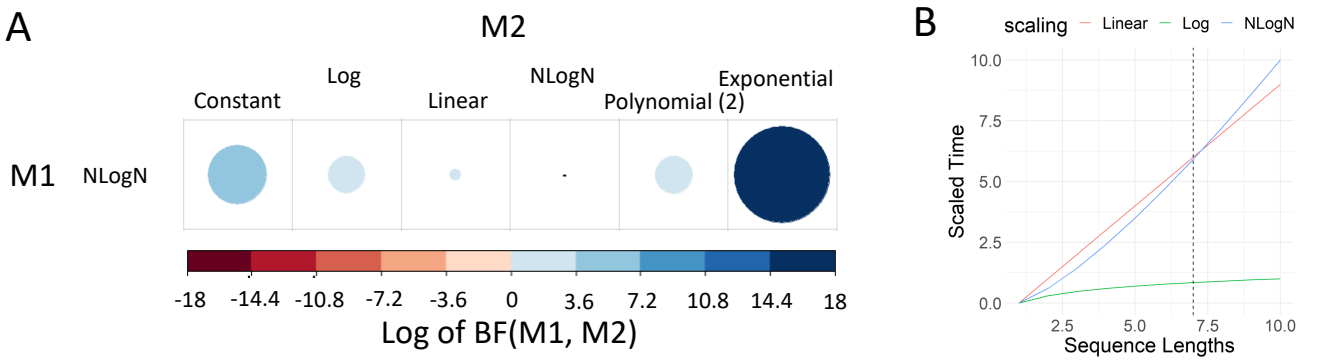

Figure E1. Scaling with $N \log N$. **A**)This plot is an Extension of Fig. 3B. We depict the log of the BFs, meaning positive values (blue) give evidence for a model and negative values (red) give evidence against it. The size and the hue of the circle represents the size of the evidence. Here the shown evidence is always the evidence for the model in which the sequence lengths (N) has been transformed according to the formula $N \log N$ (with a base of 10 for the logarithm). **B**) This plot is just a simulated illustration, that for small sequence lengths (depending on the slope and the base of the logarithm), linear and $N \log N$ scaling are very similar and potentially N*Log(N) scaling can even be favorable.