Cognition 241 (2023) 105605

Contents lists available at ScienceDirect

Cognition

journal homepage: www.elsevier.com/locate/cognit

The scaling of mental computation in a sorting task

Susanne Haridi [a,b,*], Charley M. Wu [c], Ishita Dasgupta [d], Eric Schulz [a]

[a] Max Planck Institute for Biological Cybernetics, Germany
[b] Max Planck School of Cognition, Germany
[c] University of Tübingen, Germany
[d] Princeton University, Department of Computer Science, United States of America

ARTICLE INFO

Dataset link: Mental Sorting (Original data)

Keywords:
Mental sorting
Complexity
Visual search
Structure learning
Reaction times

ABSTRACT

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 investigate 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 tested this approach in a task where participants had to sort sequences of rectangles by their size. Our results showed that reaction times scaled close to 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.

1. 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 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 constraint on 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 $\mathcal{O}$, Ω or Θ notations that track worst, best and average-case complexities respectively (Papadimitriou, 2003; Thomas H, Charles, Ronald L, Clifford, et al., 2009). We are interested in average-case complexities since in sorting several algorithms have equivalent worst-case complexity but the most efficient algorithms in practice are often selected based on differing average-case complexities (Hoare, 1962). Specifically, we focus on the average-case *time* complexity (i.e. the processing time as measured by average RT), 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

* Corresponding author at: Max Planck Institute for Biological Cybernetics, Germany.
E-mail address: susanne.haridi@maxplanckschools.de (S. Haridi).

https://doi.org/10.1016/j.cognition.2023.105605
Received 24 November 2022; Received in revised form 17 August 2023; Accepted 24 August 2023
Available online 23 September 2023

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 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' RTs, we can approximate the set of plausible algorithms underlying participants' mental computations. A similar approach to constrain the algorithms which could underlie a cognitive process has already been used by Dry, Lee, Vickers, and Hughes (2006) to investigate the scaling of how solution times in a traveling salesman problem depend on the number of nodes.

In fact, a great deal of cognitive psychology can be understood as attempting to constrain mental computations by varying the inputs and using RTs and performance measures to understand how the mental computations change with these constraints. Consider, for example, recent work by Planton et al. (2021), where RTs were used to probe how humans compressed information in auditory and visual memory tasks. RTs have also been used to measure how long people ponder before making a decision (Ratcliff, 1978; Ratcliff & McKoon, 2008) or to study set-size effects in working memory (Sternberg, 1969). In this study, we are building on the rich tradition of linking mental computations and RT by formally looking at how RTs can approximate the complexity of potential algorithms underlying a mental computation.

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. Furthermore, there is some work arguing that sorting algorithms (often in the form of stacking cups) are among the earliest algorithms people acquire and that the hierarchical organization of elements might be related to language development (Greenfield, 1991; Greenfield, Nelson, & Saltzman, 1972). 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 an average complexity of $\Theta(N \log N)$, i.e. scale super-linearly but still favorably (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 in which people had to determine whether two images showed the same object, just with a different rotation (Shepard & Metzler, 1971). RTs 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 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.

1.1. Goals

Consequently, the aim of this study is to 1. investigate how human sorting scales when more items need to be sorted and 2. 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, participants had to mentally sort sequences of rectangles of different sizes and colors by their size. To measure how their sorting time scaled, 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 exploited latent structure to improve the time complexity for mental sorting. Our results showed that for the input range we presented, participants' RTs scaled approximately linearly with the number of rectangles (though we cannot rule out scaling laws that behave very similarly to linear in our input range, like $\Theta(N \log N)$) and that they exploited the latent structures to reduce their RTs. 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.

2. Methods

To investigate the scaling of mental computations, we studied how the time people needed to mentally sort sequences scaled with increasing sequence length. We also investigated if participants could detect and exploit latent structures to improve the scaling of their mental sort.

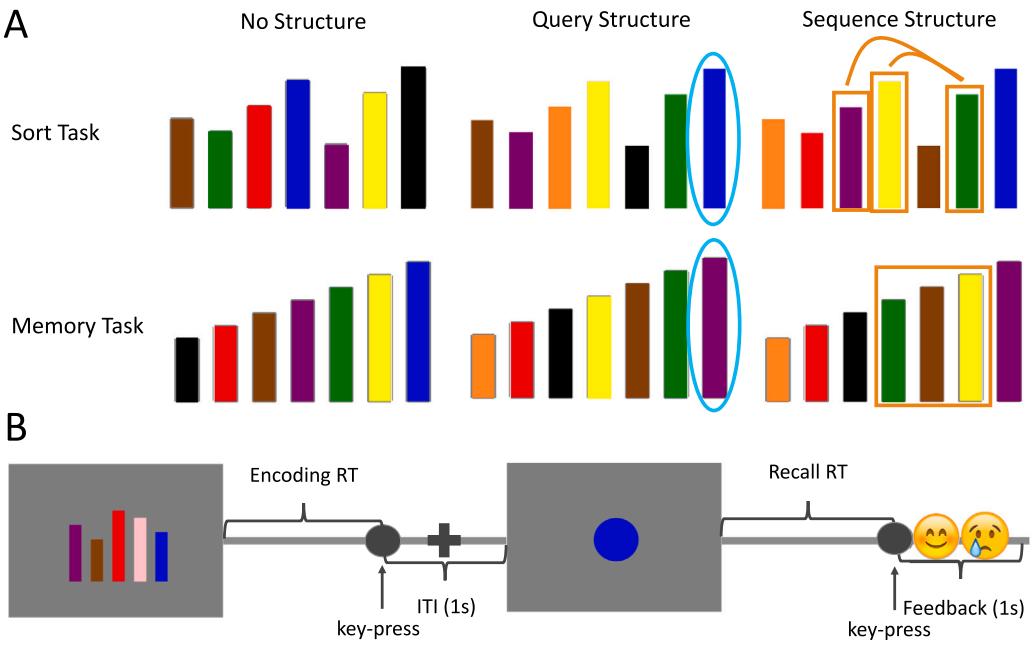

Fig. 1. Overview of the experimental design. (**A**) Schematic of the six different conditions. The sequences shown here are matched, meaning they all had the same lengths and heights and the same order for the three structure conditions in the *sort task*. (**B**) Schematic of one trial in the *sort task*. 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. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.1. Participants

We recruited 103 adults (37 female, age range: 24 to 74, Mean_{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 before the start of the experimental trials. Furthermore, we used the average accuracy a participant could achieve if they only sorted the first three rectangles (75%) as a performance cutoff. In total, 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 maximum bonus of $8.00; Mean_{reward} = $10.3; SD = $0.96; the bonus was linearly dependent on the accuracy, i.e. if a participant got 80% of the trials 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 before the experiment started. The study was approved by the ethics committee of the medical faculty of the University of Tübingen (number 701/2020BO).

2.2. 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, 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 tasks. For this purpose, all trials in the two tasks were matched by the height of the rectangles, the length of the sequences, and the queried position. To prevent any memory effect, 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 × 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 neighboring rectangles, the height of the smallest rectangle completely determined the height of all rectangles in a sequence. The *query structure* condition 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. Accordingly, the *query structure* transforms this task into a length task, which scales more favorably. Exploiting the structure of this condition means realizing that the task has changed to an easier task. 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 the other two conditions. If participants learned 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. For sequences of length three or below no sorting was necessary 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. Participants 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*. Participants were then 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 length) in each block, resulting in 210 trials for each participant. At the end of the task, participants performed a short color blindness 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. No participants were excluded due to the color blindness test.

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 1 s. 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). 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 from 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 15,330 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 s (1216 trials), to avoid including trials, where the participant had left the screen (see Fig. C1a). This left us with 13,444 trials in total.

3. Results

3.1. 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. In this study, we focused on the RTs, but it is important to note that people did make mistakes and that our manipulations influence the number of mistakes (see Fig. C1 and appendix C). The implications of this for our results are discussed in the discussion.

In the following, we first investigate the scaling of mental sorting via linear regression models. We then looked at the effects of structure on the RT and modeled how this structure could potentially be learned.

3.2. Behavioral results

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 where the predictor 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 sequence length[1] 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 contain 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, then 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, 2017, 2018) 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 the data is twice as likely under the alternative hypothesis. Generally, BFs that are larger than 3 are interpreted as giving substantial evidence for

[1] The task and structure conditions were both encoded as nominal variables, while the sequence length was numerical.

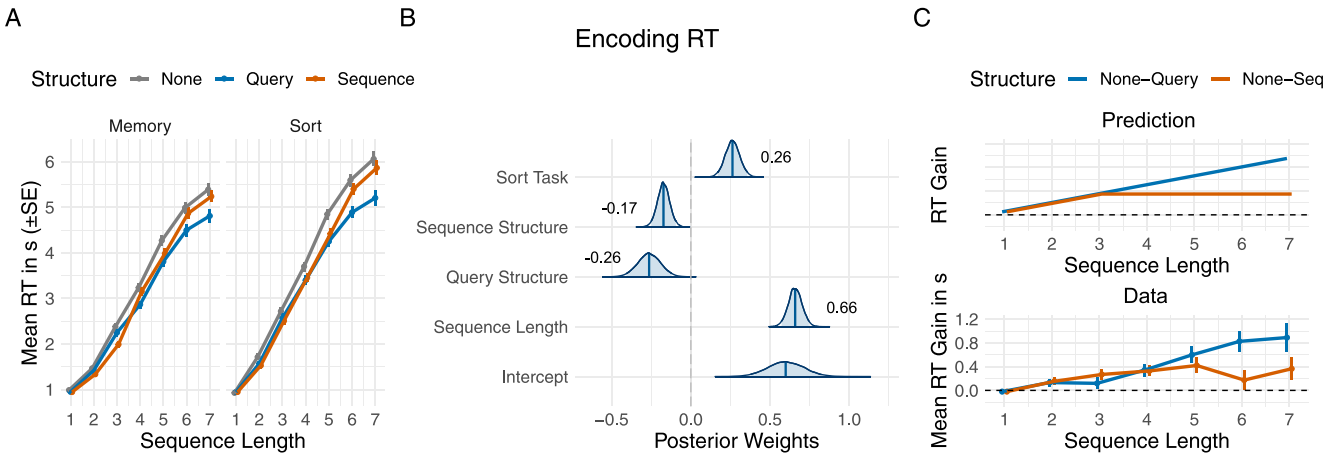

Fig. 2. Behavioral results. (**A**) Average encoding RT over all trials. The left plot only shows the trials from the *memory task*, while the right plot shows the trials from the *sort task*. (**B**) Estimates of the fixed effects of the full encoding RT model. The depicted numbers are the mean estimated 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 extent. The exact shape of the increasing gain depends on the way that mental sorting scales. The depicted linear increase is therefore just for illustrative purposes. The lower plot 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). All error-bars represent the standard error (SE). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

one hypothesis over the other. All models we used to estimate effects are linear regression models. 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 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 $\hat{R}$ values were equal to 1.

Sequence length increases RTs, while latent structures reduce RTs

Our analysis of encoding RTs showed that the full model[2] performed better than the model without the *structure* conditions[3] ($BF > 100$), the sequence length[4] ($BF > 100$), or the *task* conditions[5] ($BF > 100$) as fixed effects. The resulting parameter estimates of the full model showed that RTs increased for longer sequences ($\hat{\beta} = 0.66$, 95% HDI = [0.57, 0.82]), confirming our first hypothesis.

RTs also increased in the *sort* compared to the *memory task* ($\hat{\beta} = 0.26$, 95% HDI = [0.17, 0.36]). Including interaction effects in the model[6] revealed that this effect was mainly driven by an interaction between sequence length and task (see Table D1 for more details).

In line with our second hypothesis, we found that 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).

To make sure that the observed effect of structure was not just due to block order effects, we included the block number as an additional fixed effect in the model.[7] 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.

Next, we investigated the possibility that parts of people's mental sort happened during recall. In particular, we are concerned that participants flexibly allocate varying amounts of their sort time to either the encoding or the recall phase. If this was the case, by just analyzing the encoding RT we could be missing increases or decreases that might only be present in the recall RT. To investigate this, we analyzed the trade-off between encoding RTs and recall RTs. If people occasionally allocated larger parts of their sort to the recall phase, 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. Even when accounting for different sequence lengths or structures, this relationship remained positive for almost all scenarios (see Fig. A1B).

To further ensure that we are not neglecting parts of the sort time by focusing our analysis on the encoding RT, we also ran a full model with all RTs (encoding and recall), with the RT-type as an interaction

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

Predictors	*Encoding RT*	
	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. The estimates refer to the mean posterior estimate.

[2] RT ~ Sequence Length + Structure + Task + (Sequence Length + Structure + Task + Block | Subject)

[3] RT ~ Sequence Length + Task + (Sequence Length + Structure + Task + Block | Subject)

[4] RT ~ Structure + Task + (Sequence Length + Structure + Task + Block | Subject)

[5] RT ~ Sequence Length + Structure + (Sequence Length + Structure + Task + Block | Subject)

[6] RT ~ Sequence Length * (Structure + Task) + (Sequence Length * (Structure + Task) + Block | Subject)

[7] RT ~ Sequence Length + Structure + Task + Block + (Sequence Length + Structure + Task + Block | Subject)

effect.[8] For the same reason, we also analyzed the sum of recall and encoding RT. Neither of these analyses changed the results qualitatively compared to the results from the encoding RT analysis (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.

3.3. Scaling analysis

Having shown that there was a measurable difference between the *sort task* and the *memory task*, we investigated how sorting times scaled with increasing input size by analyzing the difference between these two tasks (see Fig. 3A). For the following analysis (unless otherwise stated), we used this difference (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 10 s 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 approximately 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 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 model representing $\Theta(Nlog(N))$ scaling, 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: $f_{lin}(s) = s$; NlogN: $f_{NlogN}(s) = slog_{10}(s)$; polynomial(2): $f_{poli}(s) = s^2$; exponential: $f_{exp}(s) = e^s$. We then did a model comparison by calculating the BFs of all pairwise model-combinations[9] (see Fig. 3B for a depiction of all results). The linear model (Linear vs. Constant: $BF > 100$; Linear vs. Log: $BF > 100$; Linear vs. Polynomial(2): $BF > 100$; Linear vs. Exponential: $BF > 100$) and the NlogN model (NlogN vs. Constant: $BF > 100$; NlogN vs. Log: $BF > 100$; NlogN vs. Polynomial(2): $BF > 100$; NlogN vs. Exponential: $BF > 100$) were better than all others, supporting that participants' sorting times scaled favorably. A direct comparison between the linear and the NlogN model resulted in evidence slightly in favor of the NlogN model (NlogN vs. Linear: $BF = 1.92$). However, it is important to note that in the space of 1–7 $f_{NlogN}(s)$ behaves very similarly to $f_{lin}(s)$ (see Fig. E1), making it impossible to clearly distinguish between these two complexities for the given input range. Nevertheless, this comparison is crucial, because ideal comparison-based sorting algorithms have a complexity of $\Theta(N\log(N))$. Our results suggest an approximately linear scaling of mental sorting in the given input range but leave open the possibility for ideal comparison-based sorting.

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 participant's 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.[10] should not improve the model fit compared to an intercept-only model[11] 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. Furthermore, to test our hypothesis that the scaling for the structure conditions should be better, we calculated a separate model for each structure. For all structures, the constant model performed better than the model containing n (*no structure* condition: $BF = 7.54$, $\hat{\beta} = 0.17$, 95% HDI = [0.01, 0.32], *query structure* condition: $BF = 9.67$, $\hat{\beta} = 0.16$, 95% HDI = [0.02, 0.3], and *sequence structure* condition: $BF = 7.31$, $\hat{\beta} = 0.17$, 95% HDI = [0.02, 0.32]), meaning we found evidence for linear scaling. The fact, that the intercept estimates of the constant model did not overlap with 0 suggests no constant scaling, supporting the conclusion of the above analysis.

To summarize, we found evidence that mental sorting scaled approximately linearly for the given input range. We managed to rule out that in the given input space sub-linear complexities such as constant or logarithmic components and super-linear complexities such as polynomial and exponential components govern the scaling of mental sorting.

3.4. 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 behavioral results already showed that participants used the latent structures to improve their RTs (hypothesis two). 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[6]. We found an interaction between *query structure* and sequence length, resulting in larger RT decreases for longer sequences ($\hat{\beta} = -0.16$, 95% HDI = [−0.24, −0.08]). For the *sequence structure* there seemed to be a small effect in the same direction ($\hat{\beta} = -0.04$, 95% HDI = [−0.08, −0.00]), but the results were less clear (see Table D1 for all estimates). 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

[8] RT ~ (Sequence Length + Structure + Task) * RT-type + ((Sequence Length + Structure + Task + Block) * RT-type | Subject)

[9] General model structure: Sorting Time ~ f(s) + Structure + (f(s) + Structure | Subject)

[10] diff ~ n + (n | Subject)

[11] diff ~ 1 + (n | Subject)

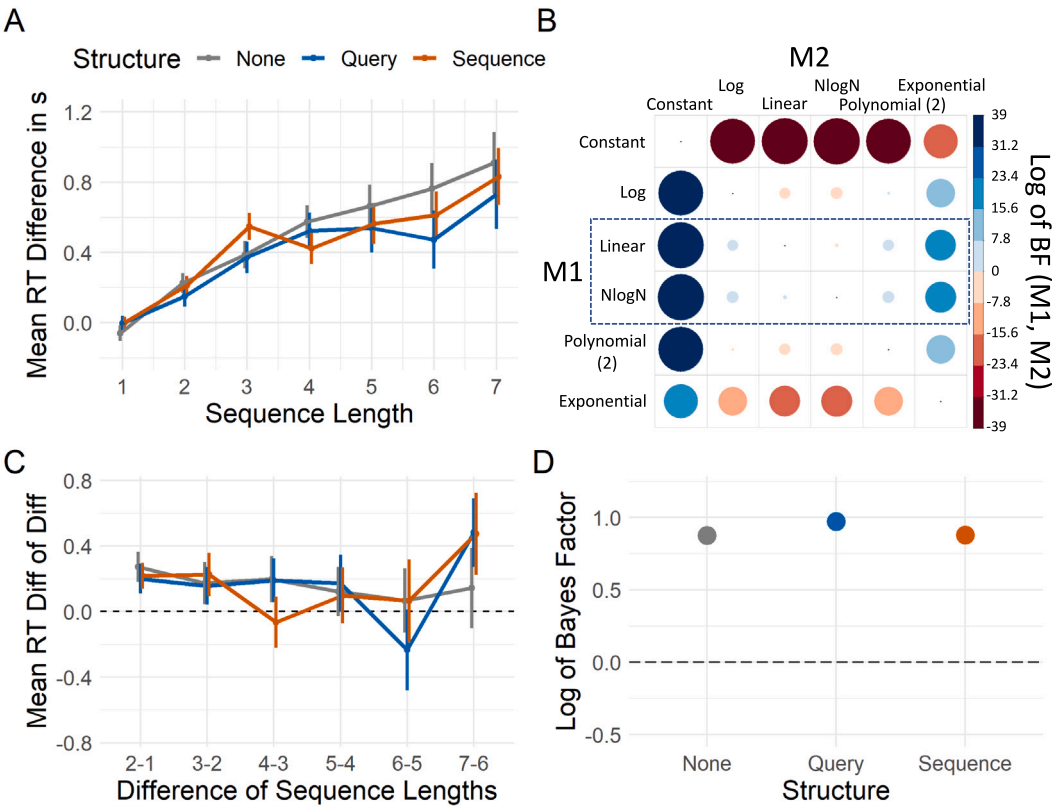

Fig. 3. Investigation of the scaling. (**A**) The *y*-axis shows the differences of the *sort task* RTs and the *memory task* RTs. This difference represents the sorting time (without the memory component). (**B**) Scaling analysis. We calculated maximally-structured mixed effects models on the RT difference depicted in **A** with the transformed sequence length as a predictor. 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 represent the size of the evidence. The rows represent the models for which the evidence is gathered, meaning that the winning model is the model where the whole row has values above zero. (**C**) Sorting times increase for each sequence length increase. The plot shows the mean of the difference in 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 favor 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 BFs here are log-transformed (as in **B**) and represent the evidence in favor of linear scaling. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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[12] (representing the hypothesis that there was no or a constant difference), the second model had the sequence length as a predictor[13] (representing our hypotheses about the benefit of the *query structure* condition) and the third model also had the sequence length as a predictor, but recoded sequence lengths above 3 as 3[14] (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 comparatively small BFs suggest that people did not benefit as much from the *sequence structure* as we expected.

In summary, as we proposed in our third hypothesis, *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.

3.5. 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 approximately 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 an average scaling of $\Theta(N)$ in exchange for being prone to 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/position (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

[12] diff ~ 1 + (Sequence Length | Subject)
[13] diff ~ Sequence Length + (Sequence Length | Subject)
[14] diff ~ re-coded Sequence Length + (re-coded Sequence Length | Subject)

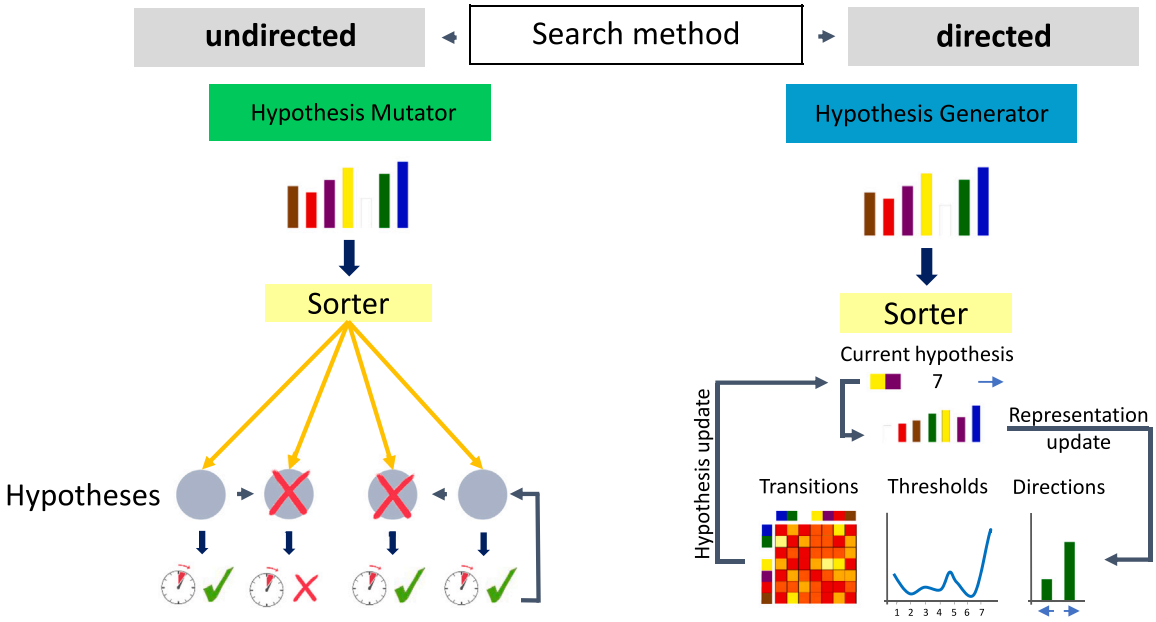

Fig. 4. Sorting models. The division illustrates the dimension on which the models differ. The illustration below depicts a schematic of the two different models. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

sort length of three and a sort that starts 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. For both models, the updating of the hypotheses depends on the feedback after the recall phase and is therefore unlikely to be captured by our RT measures. Therefore, we did not use the complexity of the structure learning process as a selection criterion. However, we do talk about the implications of differences in complexity in the discussion.

3.5.1. 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, which evaluates a limited set of hypotheses about the structure, exchanging bad hypotheses (i.e. hypotheses that resulted in wrong or correct, but slow responses) with mutated variants of better performing hypotheses. Meaning this model has two hyperparameters. 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.

3.5.2. 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 b, and the best sort direction 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 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 d encoded the sort direction based on the relative position that was queried. If the second position was queried in a sequence of length 4, then 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 on the implementation of the models, see appendix B.

Model 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 models observed the 35 trials from each block (three blocks per participant, one for each structure) in the same order as the participants. At the beginning of the 35 trials, the models always started naively. For the hypothesis mutator, this means that the model was initiated with a random set of hypotheses. For the hypothesis generator, this means that the model assumed there was no structure (i.e. the model assumed that no colors co-occur, the direction of the sort is irrelevant, and the whole sort is required). The models then sorted each trial, based on their current belief about the structure. This sort generated model times, which were defined as the number of steps the sorting algorithm executed until the sort was stopped. After receiving feedback for each trial, the model adjusted its belief about the structure according to said feedback, the model times, and (in the case of the hypothesis generator) the sequence it observed. Accordingly, the models learned the structure online while observing the same trials as the participants did (see Fig. 5D and E for the final hypotheses that the hypothesis generator learned at the end of the 35 trials for each participant). For the model comparison, we ran a Bayesian regression model on participants' encoding RTs of the *sort* task using the model times (as well as the structure) as fixed and random effects.[15] We calculated the loo R^2 values to compare the two models.

[15] RT ~ model times + Structure + (model times + Structure | Subject)

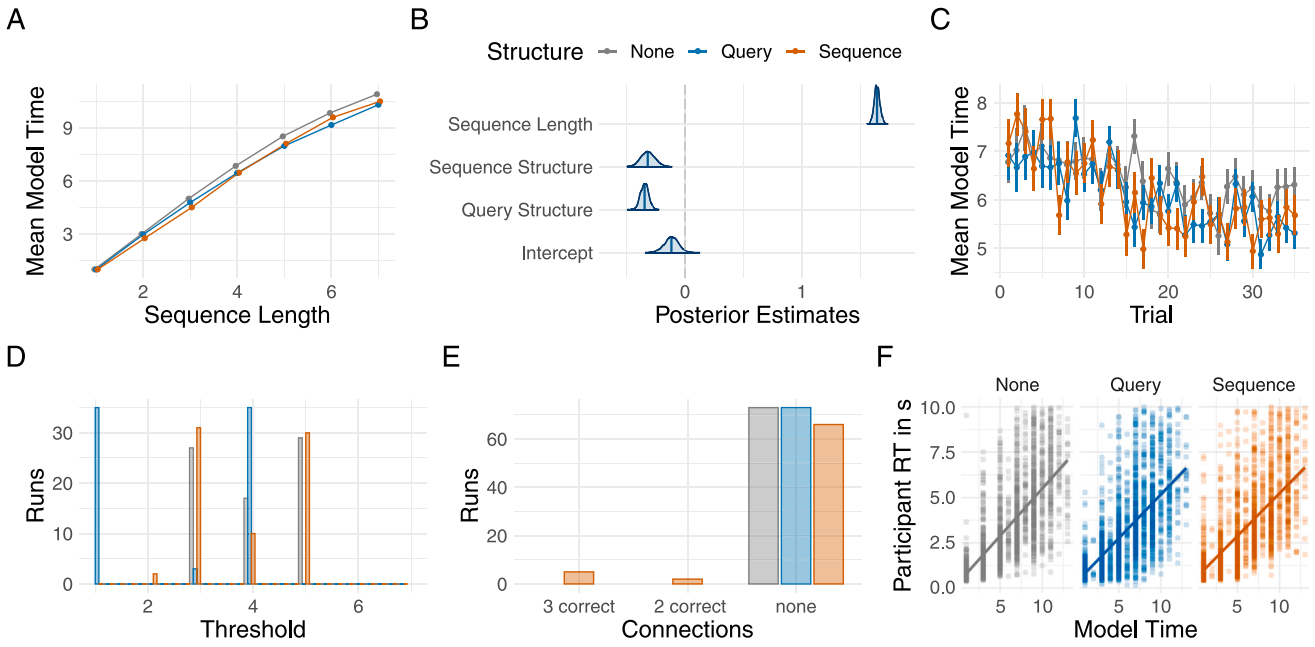

Fig. 5. Output of hypothesis generator. (**A**) This plot shows the mean model time for each sequence length over all trials. (**B**) Behavioral pattern. With this model, we investigated whether the times of the hypothesis generator have a similar pattern to the human RT data depicted in Fig. 2B. (**C**) This plot shows the improvement of the model times over the 35 trials of a block (see Fig. F1 for equivalent analysis for the human data). (**D**) This plot shows the learned thresholds, i.e. how many rectangles the hypothesis generator was willing to sort for the last trial of each block. (**E**) This plot shows whether the hypothesis generator learned the correct connections for each of the blocks. As can be seen, no wrong connections were learned. (**F**) The plot shows the relationships of the model and the real RT data for each trial. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The hypothesis generator explained the most variance in the data (hypothesis generator: loo R^2 = .59, hypothesis mutator: loo R^2 = .57). To ensure that this small difference in the loo R^2 values was meaningful, we also compared the models directly via BFs. This comparison showed that the hypothesis generator described trial-by-trial variations in the participant's RTs better than the hypothesis mutator ($BF > 100$).

Model output

To make sure that the hypothesis generator is a valid description of human behavior, we qualitatively compared the model times we 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, 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 model of participants' behavior.

Interestingly, the hypothesis generator even improved its processing time in the *no structure condition*. Specifically, the model 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, see Fig. 5D, and the improved processing times, 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.

4. 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 studied the scaling of mental computations to 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 approximately 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 modeling to compare two models that used undirected or directed search methods to learn hypotheses about the latent structure. We found that the model that applied directed search to generate and test hypotheses could replicate our observed behavioral patterns. These results show that people deal well with increasing complexities (at least at the scale presented in our experiment) and emphasize the usefulness to study how mental computations scale more generally.

One limitation of the current study is the length of the considered sequences. Due to the nature of the task, we were limited to a length that could still be maintained in working memory. Accordingly, drawing strong conclusions about the complexity of mental sorting from this limited range of input sizes is not easily possible. For instance, when compared to a linear complexity, a complexity of $N \log N$ can result in equally fast or even faster sorting times for short sequences and is therefore hard to distinguish from linear scaling. Nonetheless, we believe that with our current paradigm, we can make inferences about the super- or sub-linearity of the scaling and we managed to exclude the possibility that very favorable scaling factors (such as constant or logarithmic complexities) or very unfavorable scaling factors (such a polynomial or exponential complexities) govern the scaling of mental sorting in the investigated input space.

A related concern is that at small input sizes, a linear component might drive the scaling effects. For example, both $[2000 * n + (1/1000) *$

n^2] and [n^2] scale exponentially, but the former could explain linear scaling of the RTs. In other words, by only observing the complexity for sorting 1 to 7 rectangles we cannot exclude the possibility that the approximately linear scaling we observe just marks the beginning of an exponential or logarithmic curve. However, we argue that even conclusions about the scaling in the limited space we observe allow us to narrow down the possible underlying sorting algorithms. Furthermore, it is possible that the sorting algorithms used by humans change depending on the length of the sequences and the scaling could therefore differ 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). 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 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 a reflection of an increased difficulty to encode or recall the correct order of the rectangles for longer sequences. This scenario would only affect the memory component, but not the sort and should therefore not affect the presented analysis. This explanation is supported by the fact that errors also increased in the *memory task*. 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 mistakes would still be limited. As such, participants might have willingly allowed 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 favorable 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, 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 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 cannot 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 mental sort.

One question that our study has not yet addressed concerns the exact algorithms used by participants to accomplish approximately linear scaling. Despite eliminating a range of sorting algorithms that exhibit less favorable scaling than our results suggest, we are still confronted with numerous potential algorithms that warrant consideration. 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 like radix sort, or counting sort (Horsmalahti, 2012) could be just as likely given our current results. Another promising option would be a parallel sorting mechanism that functions like a criterion-bar that is moved either up or down all rectangles at the same time. Rectangles that 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.

One caveat here is, that humans could also adaptively allocate more neural resources to more complex tasks (Krebs, Boehler, Roberts, Song, & Woldorff, 2012; Vassena et al., 2014; Verguts, Vassena, & Silvetti, 2015). Accordingly, it is possible that the favorable increase in RTs for longer sequences is due to the parallelization of substeps of a more complex sorting algorithm. Whether the favorable scaling we observe is the result of varying neural resources or the use of a beneficial algorithm remains an open question for future research using neuroimaging techniques.

Our modeling results suggest that participants used a directed search method that was informed by the observed transitions to generate hypotheses about latent structures. The incremental hypothesis generation is 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 belief about the underlying structure. Furthermore, how 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. Furthermore, the hypothesis generator does not explicitly reward speed (as opposed to the hypothesis mutator), but nonetheless results in faster processing times for the structure conditions. In this study, we only compared two models as broad representations of a directed or undirected search across possible structures. Further studies are necessary to delineate more precise mechanisms by which latent structure can be learned in tasks like this. For instance, we only considered task-relevant structures, but it is possible that participants considered a wider variety of features than the ones we included in the model hypothesis spaces. For example, participants might choose to skip a number of elements at the beginning of a sort, to only remember the positions of the items with odd-numbered indices, or decide that sorting may be done by color rather than height. While some of these features are unlikely given our task structure, a more unconstrained hypothesis space is an important factor for further exploration in future studies. Further studies could also look at how the hypothesis learning process scales. In our current study, the hypotheses were shared across sequence lengths. This means, that the hypotheses space that is being searched is defined by the maximum sequence length, not the current sequence length. Nevertheless, the sequence length influenced the number of trial-by-trail updates. Specifically, for the hypothesis generator the updates of the transition-matrix T were dependent on the presented colors. If a color was not presented, the corresponding row of T was not updated. Accordingly, the number of updates in the matrix in any

given trial increased linearly with the sequence length. The number of updates for the threshold vector b and the 1×2 direction-vector d respectively depended on the maximum sequence length or are independent of the sequence length and are therefore constant in our task. The hypothesis mutator, on the other hand, considered a fixed number of hypotheses with random local mutations, resulting in a constant number of updates. However, since the updates of both models were performed based on the feedback in the recall phase of the experiment, they likely did not influence our RT measures, making us unable to investigate these differences in our study. Furthermore, it would be interesting to investigate how the maximum sequence length (and therefore the size of the hypothesis space) affects the time of the hypothesis learning algorithm and use that to test the plausibility of the different hypothesis learning algorithms. Theoretically, the average number of updates required by the hypothesis generator in each trial should increase quadratically with the maximum sequence length. The updates by the hypothesis mutator, on the other hand, should remain constant. However, due to the randomness of the search, the growing hypotheses space would make it harder to find valid hypotheses with this search method. This results in a trade-off between complexity increase and accuracy, which would be interesting to investigate.

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 used this approach for 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. To further arbitrate between different process-level models of mental computations, one could also combine the 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 (Anderson & Douglass, 2001) or MEG to decode their programming traces when applying a particular algorithm (Eldar, Lièvre, Dayan, & Dolan, 2020).

5. Related work

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. Thompson, van Opheusden, Sumers, and Griffiths (2022) studied how participants sorted sequences of unknown numbers, showing that several known sorting algorithms were discovered during cultural transmission chains. Sorting has also been studied using the Wisconsin Card Sorting Task (Grant & Berg, 1993) in which participants need to sort cards according to different criteria, while the experimenter changes the used criterion after the participant made 10 consecutive correct classifications. This task has not only been used to study patients with brain damage (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). And even when two problems or queries are not exactly the same, partial similarity can be leveraged (Dasgupta & Gershman, 2021). Past work has shown that this amortization of computation 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, particularly 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.

Lastly, the field of computer science has a longstanding focus on algorithm efficiency. In line with our research, Cropper and Muggleton (2019) demonstrated that it is possible to learn algorithms with minimal computational cost using a few examples. This suggests that humans may also employ similar mechanisms to discover efficient algorithms. However, their method is intended to learn any logic program. In our specific domain, where the task is already defined, it is more practical to investigate a narrower set of potential algorithms from the beginning.

6. 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.

CRediT authorship contribution statement

Susanne Haridi: Developed the study concept, Study design, Conducted the experiment, Performed the data analysis and interpretation, Drafted the manuscript. **Charley M. Wu:** Developed the study concept, Study design, Provided critical revisions. **Ishita Dasgupta:** Developed the study concept, Study design, Provided critical revisions. **Eric Schulz:** Developed the study concept, Study design, Performed the data analysis and interpretation, Drafted the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my data/code at the Attach File step

Mental Sorting (Original data) (github)

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT in order to improve the readability of single paragraphs or sentences. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgments

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

Funding

ES is supported by the Max Planck Society. 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.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cognition.2023.105605.

References

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

Anderson, J. R., & Douglass, S. (2001). Tower of Hanoi: Evidence for the cost of goal retrieval. *Journal Experimental Psychology: Learning, Memory, and Cognition*, *27*(6), 1331.

Ashcraft, M. H., & Battaglia, J. (1978). Cognitive arithmetic: Evidence for retrieval and decision processes in mental addition. *Journal 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 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. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. http://dx.doi.org/10.18637/jss.v080.i01.

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*, .

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.

Cropper, A., & Muggleton, S. H. (2019). Learning efficient logic programs. *Machine Learning*, *108*, 1063–1083.

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*, http://dx.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.

De Leeuw, J. R. (2015). jsPsych: A JavaScript library for creating behavioral experiments in a web browser. *Behavior Research Methods*, *47*(1), 1–12.

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.

Dry, M., Lee, M. D., Vickers, D., & Hughes, P. (2006). Human performance on visually presented traveling salesperson problems with varying numbers of nodes. *The Journal of Problem Solving*, *1*(1), 4.

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

Greenfield, P. M. (1991). Language, tools and brain: The ontogeny and phylogeny of hierarchically organized sequential behavior. *Behavioral and Brain Sciences*, *14*(4), 531–551.

Greenfield, P. M., Nelson, K., & Saltzman, E. (1972). The development of rulebound strategies for manipulating seriated cups: A parallel between action and grammar. *Cognitive Psychology*, *3*(2), 291–310.

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*. (36).

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

Hoare, C. A. R. (1962). Quicksort. *The Computer Journal*, *5*(1), 10–16. http://dx.doi.org/10.1093/comjnl/5.1.10.

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., et al. (2015). Interplay of approximate planning strategies. *Proceedings of the National Academy of Sciences*, *112*(10), 3098–3103.

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

Krebs, R. M., Boehler, C. N., Roberts, K. C., Song, A. W., & Woldorff, M. G. (2012). The involvement of the dopaminergic midbrain and cortico-striatal-thalamic circuits in the integration of reward prospect and attentional task demands. *Cerebral Cortex*, *22*(3), 607–615.

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.

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 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..

Planton, S., van Kerkoerle, T., Abbih, L., Maheu, M., Meyniel, F., Sigman, M., et al. (2021). A theory of memory for binary sequences: Evidence for a mental compression algorithm in humans. *PLoS Computational Biology*, *17*(1), Article e1008598.

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.

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

Thomas H, C., Charles, E., Ronald L, R., Clifford, S., et al. (2009). *Introduction to algorithms third edition*. Mit Press.

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.

Vassena, E., Silvetti, M., Boehler, C. N., Achten, E., Fias, W., & Verguts, T. (2014). Overlapping neural systems represent cognitive effort and reward anticipation. *PLoS One*, *9*(3), Article e91008.

Verguts, T., Vassena, E., & Silvetti, M. (2015). Adaptive effort investment in cognitive and physical tasks: A neurocomputational model. *Frontiers in Behavioral Neuroscience*, *9*, 57.

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. http://dx.doi.org/10.1038/s41562-018-0467-4.

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