Parsing heuristic and forward search in first-graders game-play behavior

Luciano Paz
Laboratory of Integrative Neuroscience, Physics Department, FCEyN UBA and IFIBA, CONICET; Pabellón 1, Ciudad Universitaria C1428EGA Buenos Aires, Argentina. lpaz@df.uba.ar

Andrea P. Goldin
Laboratory of Integrative Neuroscience, Physics Department, FCEyN UBA and IFIBA, CONICET; Pabellón 1, Ciudad Universitaria C1428EGA Buenos Aires, Argentina.
Torcuato Di Tella University, Av. Figueroa Alcorta 7350 (C1428BCW), Buenos Aires, Argentina

Carlos Diuk
Department of Psychology and Princeton Neuroscience Institute, Princeton University, Princeton, NJ 08540, USA

Mariano Sigman
Laboratory of Integrative Neuroscience, Physics Department, FCEyN UBA and IFIBA, CONICET; Pabellón 1, Ciudad Universitaria C1428EGA Buenos Aires, Argentina.
Torcuato Di Tella University, Av. Figueroa Alcorta 7350 (C1428BCW), Buenos Aires, Argentina

Seventy-three children between 6 and 7 years of age were presented with a problem having ambiguous subgoal ordering. Performance in this task showed reliable fingerprints: 1) a non-monotonic dependence of performance as a function of the distance between the beginning and the end-states of the problem, 2) very high levels of performance when the first move was correct and 3) states in which accuracy of the first move was significantly below chance. These features are consistent with a non-Markov planning agent, with an inherently inertial decision process, and that uses heuristics and partial problem knowledge to plan its actions. We applied a statistical framework to fit and test the quality of a proposed planning model (Monte-Carlo Tree Search (MCTS)). Our framework allows us to parse out independent contributions to problem-solving based on the construction of the value function and on general mechanisms of the search process in the tree of solutions. We show that the latter are correlated with children's performance on an independent measure of planning, while the former is highly domain specific.

Keywords: Heuristics; Stochastic Behavior Modeling; Monte-Carlo Tree Search; Planning; Children Problem Solving; N-Puzzles; First Graders

Corresponding author: Luciano Paz (lpaz@df.uba.ar)
The authors wish to express no conflict of interests in the publication of this work.
This research was supported by Consejo Nacional de Investigaciones Científicas y Técnicas, Ministry of Science of Argentina, Human Frontiers, and Fundación Conectar.
Mariano Sigman is sponsored by the James McDonnell Foundation 21st Century Science Initiative in Understanding Human Cognition—Scholar Award.

1 Introduction

Preschoolers are endowed with a general repertoire of problem-solving methods that only shows a modest change with age (Klahr & Robinson, 1981; McCormack & Atance, 2011). This is true even in games in which subgoals are difficult to parse (Klahr, 1985), refuting Piaget's view that without evident subgoals children would simply move randomly (Piaget, 1976).

To infer the repertoire of children's problem solving resources, Klahr and collaborators (Klahr, 1985) analyzed the trajectories of children's play in the state-space graph of a variant of an N-puzzle which he called the Dog-Cat-Mouse game (DCM) (Fig. 1). In this graph, nodes indicate game board positions and links represent legal moves between them. Trajectories could not be accounted for by a random-walker (an instantiation of Piaget's view of children moving haphazardly). Instead, the variance in the data was better accounted for by a walker that 1) avoids moving to the previously visited state (backup), 2) displays greediness (even when it is not optimal) and 3) is able to foresee the goal within a relatively short horizon of about two or three moves. These are forms of weak methods which lead to seemingly intelligent behavior in unknown domains (Newell, Shaw, & Simon, 1959; Pearl, 1984). In this work we revisit Klahr's ideas with three main novel objectives:

First, to examine the dynamics of children's performance (McCormack & Atance, 2011) throughout several sessions of play.

Second, to investigate whether children's performance can be described as a stochastic planner with partial knowledge of the game and infer the elements of planning (value function, search depth, stochasticity, backup avoidance).

Third, to identify which of the parameters of a model calculated for each child are predictive of the child's performance in a different planning game.

1.1 The Game

The data were collected during an experiment where a total of 73 low Socioeconomic Status (SES) children, 6-to-7 years old, played three different computer games during 27 non-consecutive school days (Goldin et al., 2014). Here we focus on one of the games played during this intervention; a variant of the DCM game introduced by Klahr (Klahr, 1985). This variant consists of three characters (a boy, a girl and a cat) and their corresponding homes (see Fig. 1 upper panels). The characters can only be moved along the paths into empty spaces, one at a time. A problem is defined by the distribution of characters in the places in the initial state. The goal of each game iteration is to move every character to its corresponding place. A detailed explanation can be found in Goldin et al 2013 (Goldin et al., 2013).

In the state-space graph, each game configuration is represented by a node. Links between nodes denote the existence of a legal move connecting the two represented states (Fig. 1). From this representation one can easily define a distance between any two given states as the minimal path between the two states, which corresponds to the minimal number of moves required to go from one state to the other. This state-space graph reflects the existence of two types of moves which were noted in Klahr's original study (Klahr, 1985): 1) rotations around the graph rings, which in the game correspond to actions that move the characters through the peripheral ladders (Fig. 1) and 2) permutations, which link two states of different rings, corresponding to actions that move the characters through the diagonal ladder. This representation also reveals two classes of nodes: 1) Those in which only rotations are possible (the two houses of the diagonal are occupied) and 2) those in which there are three possible moves, two rotations and one permutation (a location in the diagonal is unoccupied).

A very important aspect of this game (and, more generally, of all games which can be represented by a state-space graph) is that it has Markov transitions. This means that the consequence of actions are insensitive to the history of the game and only depend on the current state.

1.2 Analytic Strategy

The main objective of this work is to infer, from the distribution of trajectories in the state-space, the interaction between the two constituents of the planning process: the expected value assigned to each state and the search procedure that determines how states are sampled. To quantitatively asses these issues, we implement a computational model of planning and probe its ability to fit children's behavioral data.

We first analyze the performance of an Artificial Intelligence (AI) agent that uses a Monte Carlo Tree Search (MCTS) planning algorithm to select its moves. In the DCM game, the effects of making a move are deterministic and the game is fully-observable, as opposed to a partially observable game in which the player has incomplete information of the game's state, and usually many possible states can correspond to the available information. Thus in a fully observable game, the player can uniquely identify any state with the observed information. These imply that we can represent the agent as facing a Markov Decision Process (MDP) (Puterman, 1994). The game's state-graph defines the environment's possible states $\mathbb{S}$ and the legal moves are the available actions $\mathbb{A}(s)$ which change the state of the game from the original s to a different s'. It is important to emphasize that

Abbreviations: Temporal Difference (TD), Reinforcement Learning (RL), Monte-Carlo Tree Search (MCTS), Socioeconomic Status (SES), Dog-Cat-Mouse game (DCM), Markov Decision Process (MDP), Artificial Intelligence (AI), Degrees of Freedom (df)

an agent can act in a non-Markov manner (choosing actions based on the history of its moves) even if the environment has Markov transitions. In fact, this will turn out to be a major finding of our analysis.

In the next section we implement a planning model which describes children's performance as a stochastic search in the tree of possible moves. The different branches of the decision tree are not explored homogeneously. Instead, the planner opts with higher probability to explore branches that start in high valued states. This model is only meaningful if four observations are satisfied in children's behavioral data:

I. Children must know the dynamics of the game (i.e. transitions between game states). This is not true for all planning problems, where the effects of actions made by an agent may vary due to many factors (e.g. environment randomness, an opponent that executes an action that affects the problem's state, etc). Supervising adults that accompanied each child while they played the game verified that children in fact understood the actions of the game (Sec. 2.2).

II. Children's performance in DCM must be stationary, i.e. not vary considerably with increasing number of trials (analyzed in Sec. 3.1). If this were not the case, the parameters of the planner should also vary during the game through a learning mechanism.

III. Children's behavior has to vary from trial to trial and they must not select deterministically the same sequence of moves when starting from a given state (discussed in Sec. 3.2).

IV. Children's behavior must not be purely random (analyzed in Sec. 3.2).

All the above features are observed in children's data as specified in the mentioned sections.

In addition to these observations, our model has a set of general assumptions:

- Children don't consider all possible sequences of moves before acting. Hence, we assume they do a form of approximate planning
- The possible sequences of moves that children analyze while planning must be a small sample from the space of all sequences.
- In order to sample a small number of possible sequences and perform well, the sampling mechanism must be biased toward what children interpret as the most "promising" moves.

Based on these assumptions, the model which we implement can be used to examine the following set of hypotheses:

A. The stochastic search has a shallow depth of 2 moves (Klahr, 1985) (Sec. 3.3).

B. The search process is influenced by the previous moves taken (Sec. 3.2 and 3.3).

C. Planning parameters related to search resources should transfer to different problems. However, the quality of value estimates, which are expected to be domain specific predictors of performance in the game, should show no transfer to different problems.

2 Methods

2.1 Children's behavioral data

The corpus of data we used comes from a large-scale school intervention (Goldin et al., 2014, 2013; Lopez-Rosenfeld, Goldin, Lipina, Sigman, & Slezak, 2013). This study is based on data obtained from a school intervention performed in 2010 (Goldin et al., 2014). A total of 111 low SES 6-to-7-year-old children (62 males) participated in the study. All participants were recruited from five classrooms in two schools of the City of Buenos Aires, Argentina, during the second semester of their first year at school.

The intervention involved three stages. In the first stage, children performed a battery of cognitive tests, which included a physical (without using computers) implementation of Tower of London (ToL). In the second stage, children were divided into a control (38 individuals) and experimental (73 individuals) group. The control group played three games which were less cognitively demanding, while the experimental group performed three games that were designed to exercise memory, inhibitory control and planning. In the third stage, children repeated the battery of cognitive tests they had done in the first stage. All the training and testing procedures were conducted by the investigators inside the schools, in rooms appropriate for these purposes. Children's caregivers gave written consent to participate in the study, which was previously authorized by the Institutional Ethical Committee (CEMIC-CONICET). In each classroom children were balanced for gender and randomly divided into experimental and control groups.

The intervention involved a total of 27 nonconsecutive school-day sessions of 15-to-20 minutes of training on three different computer games, one of which was the variant of the DCM game presented in section 1.1. At most three experimental sessions were conducted every week. Children played three sessions of each game and then changed to a different one. That is, if children played the DCM game on Monday, Wednesday and Friday, the following Monday they would play a different game and after three sessions they would play the third one. After 9 sessions they would play the DCM again. Due to absenteeism and school issues, not every child completed the 27 sessions. On average, children

played 24 ± 1 sessions. The battery of cognitive tests was administered one week before and after the training.

This intervention resulted in a vast set of data. One of it's specific objectives - which was addressed elsewhere (Goldin et al., 2014, 2013) - was to measure whether playing these games had an impact on children's school performance. Here we capitalize on the dataset generated on children's play in the DCM game to infer planning mechanisms. We also use the fact that children played another planning game (ToL) to compare performance in this game to parameters of the model identified for each child for the DCM game.

2.2 Dog-Cat-Mouse data

Before beginning to play the DCM game children were instructed to move each character to its home according to three rules: the characters have to be moved one at a time and to an empty place (this is called a "move"), they can only be moved through the bridges, and they cannot share a house. They were also told not to rush since speed was not necessary to win. The game is implemented in Javascript and an updated version can be found at `http://www.matemarote.com.ar/`. To move a character, children click on it, then drag and drop it to the new position. If the drop is made outside of a house or if the children come back to the original position the action is not counted as a move. In all sessions, every child played accompanied by an adult who was there to explain the rules (the first time) or remind them of the rules (whenever necessary), and to support the child if needed (for instance, some children need somebody to tell them that they play well and that it is part of the game if they lose). All experimenters gave the same instructions every time they explained the rules. Each supervising adult noted that the children had understood the rules perfectly after less than three trials. In particular, children fully understood the correspondence between their actions (movements of the mouse) and consequences in the game (displacements of a character). This justifies the use of a model that does not have to actively learn the transitions between states to attempt to account for children's behavior (as stated in condition I enumerated in Sec. 1.2).

Children played the game in a sequence of four phases. In each phase, the game trial was counted as correct when children moved all the characters to their homes. Additionally, in all phases except phase 4, children were told the minimum amount of moves it took to solve the puzzle.

In phase 1, children received positive feedback when they moved all the characters to their home even if they did so in more than the minimum number of moves necessary. Instead, in phases 2, 3 and 4, children were told that they would lose if they did not solve the puzzle in the minimum number of moves and received negative feedback if they performed the minimum number of moves necessary to win and hadn't solved the puzzle (i.e. they were forced to solve each puzzle in the minimum amount of moves or else they lost on that trial).

In phases 1 and 2, children first played a trial in which the initial configuration could be solved in 2 moves. After the children won 3 consecutive trials, the distance of the initial configuration from the goal was increased by one move. This was iterated until children played three consecutive trials correctly at the maximal distance, after which they moved to the subsequent phase.

In phase 3, the distance of the initial configuration was randomized. Children advanced to the next phase after completing 6 checkpoints. They advanced one checkpoint after correctly solving three consecutive trials.

Phase 4 proceeded as phase 3 with the only difference being that children weren't told the minimum number of moves necessary to win the trial. Children played the phase 4 repeatedly until the end of the intervention (i.e. children did not go back to previous phases regardless of the number of errors made).

For the analysis of performance shown in (Fig. 2) we considered trials to be correctly solved only if they were done in the minimal number of moves.

2.3 Planning algorithm

For a given problem (for example, an instance of the DCM game), a planning algorithm identifies a sequence of actions that lead to a goal state or, more generally, maximize cumulative reward (Russell & Norvig, 1995). Classic planning algorithms achieve this by performing forward-search through the state space of the problem. This is done employing a recursive method that expands a tree of possible output states from the starting "root" state. When the size of the state space is too large, exhaustive search becomes unfeasible. One way of dealing with this computational barrier is to sample trajectories through the state space instead of exploring it fully, ideally using a sampling strategy that is biased towards the most promising parts of the search space. Monte Carlo Tree Search (MCTS) is a family of algorithms that employ such a strategy.

This search method guides the selection of one branch over the other using a value that ranks the states. A possible definition of the value of a state is the one used in the Markov Decision Process (MDP) framework and in Reinforcement Learning (RL) (Puterman, 1994; Sutton & Barto, 1998). Here the agent receives a reward at each state and its objective is to adapt its behavior to maximize long-term cumulative reward. The value of a state is then defined as the expected future reward the agent can obtain starting from it and following a given plan. In the type of game we consider here, we assume the agent only receives a positive reward when reaching a goal state (or set of goal states), and zero reward at every other state. By using a discount function, a

maximizing agent tries to reach the goal state in as few steps as possible.

Each state's value can be estimated by a value function (Péret & Garcia, 2004), as the average of the outcomes of simulated games (Kocsis & Szepesvári, 2006), or deduced from the structure of the game (McDermott, 1996). Additionally, some algorithms, like those in the MCTS family, back-propagate the actually observed reward to the previously visited states and "learn" truer value estimates (Kocsis & Szepesvári, 2006). Here we propose a model based on a value function which combines heuristics with partial knowledge of the game.

Many MCTS algorithms do not have knowledge of state values before searching (Browne et al., 2012). The value function is calculated during search, by performing several simulations (called *roll-outs*) of the game. These algorithms estimate the value of unknown states in a Monte Carlo fashion and then behave in a deterministic manner.

The planning algorithm we employ here has two main differences: 1) It has prior estimates of the value functions and 2) it stochastically generates several sequences of actions (plans) using a search method guided by the state. We will call each stochastic process that generates a plan a *roll-out*.

The DCM state graph is an undirected and cyclic graph so the expanded trees can be infinitely long. Our proposed model assumes a maximum plan length, H, that we call the search horizon and a fixed amount of roll-outs, R.

The value of a state that can be reached at discrete time $t+1$, $V(s_{t+1})$, is determined by:

$$\begin{array}{lcl} V(s_{t+1}|s_{t-1}) & = & \gamma^{D(s_{t+1})} - nr(s_{t+1}, s_{t-1}) \\ D(s_{t+1}) & = & \eta K(s_{t+1}) + (1-\eta) G(s_{t+1}) \\ nr(s_{t+1}, s_{t-1}) & = & \begin{cases} nr & \text{if } t \geq 1 \text{ and } s_{t+1} = s_{t-1} \\ 0 & \text{otherwise} \end{cases} \end{array} \quad (1)$$

In these equations, $D(s_{t+1})$ is an estimate of the distance between the state s_{t+1} and the goal. D derives from two different sources. First, a faithful representation K of the distance between s_{t+1} and the goal. Second, a heuristic function G (G for greediness) which simply computes the number of characters that are outside their home. Based on Klahr's results (Klahr, 1985) we assumed that children would use this heuristic to estimate the proximity to the goal.

In this model, η simultaneously controls two different aspects of children knowledge. For values of $\eta \simeq 0$, the behavior of the model is determined by greediness, reflecting no internal knowledge of the game. When $\eta \simeq 1$ the value function reflects a perfect internal knowledge of the distances in the game. This means that η controls whether value is guided by an heuristic of greediness or by a correct representation of the game structure.

We also run a model with an additional parameter, σ, which can independently control: 1) the relative contributions of the heuristic and of the correct representation to the value function, and 2) an added internal noise to the value function. Noise was added as follows:

$$\begin{array}{l} \hat{V}(s_{t+1}|s_{t-1}) = \gamma^{D(s_{t+1})} - nr(s_{t+1}, s_{t-1}) + n \\ n \sim \mathrm{N}(0, \sigma) \end{array} \quad (2)$$

where n is a random number sampled from a gaussian probability distribution with zero mean and σ standard deviation. In this stochastic modification, high σ gives completely random value estimates and leads to random action selection whereas $\sigma \ll 1$ leads to a stable internal representation of the value.

Here we make explicit certain assumptions and limitations of our model. First, we have largely simplified the problem by assuming that children have some degree of knowledge of the game K and are guided by a heuristic of greediness G. This has several limitations: first there are other domain specific heuristics that are not incorporated in this simplified model. For instance, children have a preference to rotate characters clockwise (see Fig. 4.a).

Second, here we do not provide a general mechanism (that is, a mechanism which is not problem specific) to produce these heuristics (Geffner, 2010; Hoffmann & Nebel, 2001). For instance, relaxation, a mechanism by which specific constraints (or rules) of the game are ignored (relaxed) constitutes a general procedure to produce heuristics. In our specific case, G can be obtained by relaxing the rules that characters cannot overlap in one home and that characters ought to move through the ladders. Third, we do not provide a mechanism by which children may obtain the knowledge of the internal structure of the game (i.e. how they can compute K).

Our approach instead relies on deriving a heuristic which incorporates known resources from previous work (Klahr, 1985) and imperfect knowledge of the game which is represented in K. The agent with values of $\sigma = 0$ and $\eta = 1$ will have access to perfect knowledge (although play will not be perfect due to noise in the search process). As σ increases and η decreases, the knowledge of the game worsens and is driven by noise or purely greedy behavior. In our model we can fit, for each child, the relative weight of these parameters for this fixed class of hand-coded heuristics, and explore how they form the value estimate.

The parameter γ maps the distance estimate $D(s_{t+1})$ to a value estimate. This is an implementation of a delayed reward exponential decay analogous to the reward discount in the RL and MDP frameworks (Sutton & Barto, 1998) and for which a biophysical correlate has been described (Roesch, Calu, & Schoenbaum, 2007).

The $nr(s_{t+1}, s_{t-1})$ function is a non-Markov function that penalizes the value of back-up moves (Klahr, 1985) [1]. As

[1] Technically, a probability distribution has the Markov property (leads to sequence of states forming an order 1 Markov chain) when

the parameter nr grows larger, the action that produces the back-up becomes less valuable.

The search process starts from a given state s_0 at time step $t = 0$ and generates R sequences of H consecutive moves. Since moves in the game have a deterministic effect, a sequence of moves is equivalent to a sequence of output states given by each move. The probability of selecting an output state s_{t+1} during each roll-out was taken from a softmax probability distribution:

$$P(s_{t+1}|\{s_t, s_{t-1}\}) = \frac{e^{\beta V(s_{t+1}|s_{t-1})}}{\sum_{\text{reachable } s'_{t+1} \text{ from } s_t} e^{\beta V(s'_{t+1}|s_{t-1})}} \qquad (3)$$

The β parameter is called the softmax inverse temperature. When it's high, the probability of selecting the highest valued state s_{t+1} is almost 1 and when it's low the distribution becomes uniformly random. After having R stochastically generated plans, the model selects the one that has the largest end-state value, $V(s_H|s_{H-2})$, and executes it. If there are multiple plans that have equally valued end-states, it chooses randomly amongst them. If the goal is reached during the executed plan, the consecutive moves do nothing and the end value is the value of having reached the goal. If the goal is not reached during the plan, the search process restarts from s_H and takes into account s_{H-1} and s_{H-2} to recalculate the value function.

To clarify how the algorithm works, we show the pseudocode of the plan selection (algorithm 1) and summarize the model's parameters in Table 1.

Our model stochastically travels through the game graph but it does not do so blindly. It tends to follow the best possible output given prior value estimations. For instance, if the model has a large nr constant, the value of the backup moves is greatly penalized and thus they are less likely to be selected. Another feature is that when the model has a high β, the model tends to always select its believed best output state. If β is very low, it selects almost randomly. A perfectly performing model would be one with $\eta = 1$, a very high β and $\gamma \in (0, 1)$. For the extended model with noise in the value estimate, the perfectly performing parameters would be the above in addition to adding $\sigma = 0$.

The model is capable of parsing whether children's imperfect performance results from imprecisions in the value function or noise in the search process. For instance, a highly deterministic search process based on an imprecise value function will result in recurrent and stereotyped errors corresponding to local maximums of the value function. Instead, an accurate value function sampled with a very noisy search process would yield to a non-structured pattern of errors.

The planning model parameters are the search horizon H, the number of plans constructed during the search process R, the softmax inverse temperature β, the reward discount γ and the value function weights η and nr. In this work, we fit the last four parameters for $R = 3$ and $H = 1$, 2 or 3 so as to minimize the squared difference between each child's and the model's mean performance vs distance curve. We also do this setting $nr = 0$ and thus not including inertia. We simulate the model's mean performance in the corresponding child's played states, repeat the simulation 10 times, and take the mean vs distance. We consider this to be the model's mean performance vs distance data which we use to calculate the squared difference with the children's data. We then use MATLAB®'s *fmincon* function with the interior point algorithm to find the set of parameters that minimize the squared difference between each child's and the model's mean performance vs distance data. The parameter values were searched in the intervals $\beta \in (0, 20]$, $\gamma \in (0, 1]$ and $\{\eta, nr\} \in [0, 1]$

The fitted models correspond to Markov (when $nr = 0$) and non-Markov agents ($nr \neq 0$) that attempt to reach the same performance as each child. In the Results section we show how well the model fits the data and conclude that the children are inherently non-Markov.

Additionally, we fit the extended model parameters, which include σ, for a range of β and σ values in order to show that the non extended model is robust against noisy value representations.

3 Results

The results section is organized as follows: in Secs. 3.1 and 3.2) we show that the validity of the behavioral observations (stationary performance, II, and stochastic, III, but not fully random behavior, IV) which justified the choice of the model.

In Secs. 3.3 and 3.4) we assess the planning model's capacity to fit the data and test the working hypothesis enumerated in Sec. 1.2 (A, B and C).

3.1 Evolution of children's performance throughout multiple sessions

We first investigate how children's performance changes over time. In the first 9 trials[2] performance was high, $(85 \pm 1)\%$ (Fig. 2). As the game progressed, children faced initial conditions which were further from the target, as described in section 2.1. We observed a consistent drop in performance, found in around 70% of the children in specific trials.

it conditionally depends only on the current state. As we will use the value estimates in 1 to set a probability distribution, we say that function nr is non-Markov as it breaks this conditional dependence.

[2]During these the children won if they reached the goal state in less than 50 moves, but in this analysis we considered a trial to be correctly solved (performance=1) if the goal state was reached in the minimum number of moves.

```
   Input: starting state s_0, goal state G, H, R, γ, η, β and nr
1  Main Procedure: begin
2      t ← 0;
3      s_t ← s_0;
4      s_{t-1} ← s_0;
5      while s_t ≠ G do
6          Selected_plan ←rolloutPlan (s_t,s_{t-1});
7          Execute the actions in Selected_plan and
           update s_t, s_{t-1};
8          t ← t + (Number of actions in Selected_plan);
9      end
10 end

11 Function rolloutPlan (s_t,s_{t-1}): begin
12     Selected_plan ←Initiate as an array of H moves;
13     plans ←Initiate as a list of R empty arrays of H
       moves;
14     for i ← 1 to R do
15         single_plan ←Initiate as an array of H moves;
16         for t' ← 1 to H do
17             a' ←find executable moves from s_t;
18             s' ←states reachable from s_t;
19             V' ←compute values of s' with
               Value(s',s_{t-1});
20             s_{t-1} ← s_t;
21             s_t ←Select one state from s' using softmax
               probability with values V';
22             plans [i][t'] ←Assign the action that
               produced the selected s' state;
23             if The selected s_t == G then
24                 break;
25             end
26         end
27     end
28     Selected_plan ←Select the plan in plans with the
       largest end-state value;
29     return Selected_plan;
30 end
```

Algorithm 1: This shows the pseudocode of the planning algorithm. For the sake of clarity it is divided into the main procedure which "plays" the game and the function which produces the roll-out plans and selects amongst them. The required inputs are the starting and goal states, the planning horizon H, the amount of search roll-outs R and the parameters γ, η, β and nr. In the case of the extended model, parameter σ must also be added. The first lines in the main procedure initiate the time-step t, the current state s_t and the last visited state s_{t-1}. At line 5, the game starts and the planning process executes until the goal is reached. At line 14, it starts to generate the R independent roll-outs. The search for the sequence of H moves to execute begins in line 16. It is guided by the values of the possible output states s' computed in line 19. Once all R plans are generated, they are compared (line 28) and only one is selected (line 29). The selected plan is then executed (line 7) and the time step is increased. Once this is done, if the goal has been reached, the algorithm stops, if it was not, the algorithm restarts at line 5.

This decrease was modest but significant - a t-test comparing the last 200 trials' mean performance against the performance data in windows of 30 trials revealed a significant drop only between trials 30 and 90 ($p < 0.0009$ $df = 29$, Bonferroni corrected (Dunn, 1961)). This drop is observed when the initial configuration of the game is set at distance four from the goal, and performance slowly rises over a large period of trials when game is set at distance 5 from the goal (Fig. 2). Beyond this initial drop, performance remained relatively stable and stationary. A raster plot, displaying the performance of each child throughout the experiment, shows that children persist in doing a relatively small fraction of mistakes throughout the experiment.

The increase in the variance of the mean performance toward the end of the experiment results from the fact that fewer children contribute to the mean in this stage.

Based on these observations we conclude that the assumption of stationary performance corresponding to condition II of Sec. 1.2 is consistent with the data.

3.2 Children's performance in the state-space

Since children's mean performance over time remains approximately constant, we pool together the data from all 4 phases of the game in order to analyze performance in the game graph. Children's selection of moves is not random. If this were the case, children would lose more often at states that were farther away from the goal. Instead, we observe that performance reaches a minimum at a distance of five to the goal and then ramps-up again for initial states closer to the antipode (i.e. the state furthest away from the goal, Fig. 3.a). This non-monotonic dependence is quantitatively confirmed by a binomial test. We assume that the number of trials won and lost at a starting distance of 4 comes from a binomial distribution. We use the observed counts to compute the binomial parameter and its 95% confidence interval. We then compute the probability of obtaining a number of wins lower or equal to the observed count at distance 5 states, from a binomial with its parameter equal to the lower bound of the distance 4 confidence interval. This gives us $p < 10^{-30}$ that the performance at distance 5 is greater or equal to the performance at distance 4. Analogously for distance 5 and 6, we perform a similar test using the higher bound of the distance 5 binomial estimate and find $p = 10^{-10}$ of distance 6 states having lower or equal performance to distance 5. This is one (among many) demonstrations that children's play is structured and far from random. On the other hand, children's play has some intrinsic stochasticity. Again, there are several demonstrations of this, but one clear example is that each child shows intermediate levels of performance at the majority of states (for example, every single child in this study showed a performance level in distance 5 states between 30 and 70 percent).

When analyzing performance for each state in the game

graph (Fig. 4.a), we observe an abrupt performance drop below the graph's equator, with a marked rise at the goal's antipode. Above and beyond the previously described dependence on distance, in this representation we can also see a left-right asymmetry in performance. An analysis of the first move of each trajectory reveals stereotyped aspects of behavior: children tend to move the characters clockwise (Fig. 4.a) and for both distance five states of the inner ring, correct selection of the first move drops significantly below chance levels ($0.35 \pm 0.05\%$ when chance is at 66%).

These results point to the fact that children's behavior is variable, but displays stereotyped patterns. Combined with the observations stated in Sec. 3.1 and the fact that children understood the rules of the DCM game and the possible state transitions, the four necessary hypotheses that are required in order for the proposed planning model to be meaningful (I, II, III and IV in Sec. 1.2) are satisfied.

Another very important feature observable in the data is that the movement selection policy changes dramatically depending on the history of past moves (i.e. children's movement selection policy is non-Markov).

To exemplify this we first measure the selection rate when this choice is made on the first move of the sequence. We compare this, to choices made at the same state when the move is done later in the sequence, in particular, after the first move was correct (Fig. 5). To statistically confirm that these two distribution of choices are different, we perform Pearson's χ^2 tests (Plackett, 1983) and G-tests to reject the hypothesis that both move selection rates come from the same multinomial distribution. When testing the move selection distributions for the pooled data of every child, we find significant differences for all states ($D \geq 2$), even after applying the Holm - Bonferroni correction (G-test $p < 0.047$, Pearson $p < 0.03$). These differences are most significant for every state between distance 4 and 6 ($p < 10^{-6}$ for both tests with $df = 1$ or 2 depending on the number of moves available at the state). To see that this pattern holds for each child separately, we construct contingency tables containing the summed number of correct and incorrect, first and passing after first correct moves for all states between $D = 4$ and $D = 6$. Then, we test (g-test and pearson test) if the number of correct and incorrect selections for each condition come from the same binomial distribution. We do this for children with more than 125 trials played (64 out of 73) in order to have enough data to perform powerful tests. All the tested children show significant differences between first and passing, correct and incorrect movement selections, even after Holm - Bonferroni correction ($p < 0.01$).

3.3 Model for children's performance in the state-space

As observed in Sec. 3.1, children's performance over time remains approximately constant. This allows us to simplify the game-play model by using a non-learning planning agent [3] (i.e. the agent will show a stable running mean performance), such as the one detailed in Sec. 2.3.

The planning model proposed in Sec. 2.3 uses several parameters (Table 1) to determine its gameplay. We fit values of β, γ, η and nr for each child by minimizing the squared difference between the measured and simulated mean performance as a function of distance. The number R is fixed at 3, and H was set to 1, 2 and 3, producing different parameters for each value of H. As stated in Sec. 3.2, we use all the trials played by each child. We do this in order to produce better statistical estimates, and also because for several analyses, there are not sufficient data in each phase for reliable regressions. In A we show that restricting analyses to phase 4 (the phase for which we have more data) yields performance fits of similar quality as a function of distance, when compared to the analyses that consider all phases. We also fitted the model parameters while setting $nr = 0$. We refer to this case, with $H = h$, as the model without inertia, and represent it as $\langle H{=}h, nr = 0 \rangle$. The case with inertia is represented as $\langle H{=}h, nr \neq 0 \rangle$.

The non-inertial models (Fig. 3.b, left panel) could not reproduce the children's mean performance vs distance data, and for $H = 1$ and 2, yielded monotonically dropping performance as a function of distance. The $\langle H{=}3, nr = 0 \rangle$ shows a higher performance at distance 7 (in the antipode) than performance at distance 6. The explanation for this observation resides in specific details of the state-space graph. First, note that the antipode and the target are in two different rings of the graph. Hence, simply by chance the fraction of correct n-plans (which include moving from one ring to the other) is greater when parting from the antipode. This effect is more prominent for odd depths, because the diagonal move is available only in one of every two configurations. Second, the greedy heuristic tends to bias the model towards configurations in which the majority of the pieces are in the correct place. The antipode is a greedy local minimum (one of the characters is in its home position). Hence, the agent (driven by greediness) often returns to the antipode when it computes two moves forward. When it computes three moves forward ($H = 3$), it is impossible to generate a plan that starts and ends at the antipode. The end-state value is not as biased in this case, and it is more likely to escape from the local minimum. We perform a Pearson χ^2 statistical test (Plackett, 1983) to test if the different models and the children's total number of wins significantly differ. They all do with $p < 10^{-6}$ for $H = 1$ and 2 and $p < 0.01$ for $H = 3$ all with $df = 1$. Hence, from these results we can conclude that model fits without inertia are both qualitatively and quantitatively inaccurate.

The inertial models (Fig. 3.b, right panel) $\langle H{=}1, nr \neq 0 \rangle$

[3] In the present framework, we say that AI agents learn if they update the values of the states according to their performance in the game, in a way that makes them play better.

and $\langle H{=}3, nr \neq 0\rangle$ also yielded poor quality fits and significantly differ from the children's mean performance ($p < 10^{-6}$, $df = 1$). Only model $\langle H{=}2, nr \neq 0\rangle$ does not significantly differ ($p = 0.95$, df=1). Thus $\langle H{=}2, nr \neq 0\rangle$ is the best of all the candidate models to describe the children's DCM gameplay data. The fact that the best fit occurs when $H = 2$ is consistent with our working hypothesis A based on Klahr's data (Klahr, 1985).

The inertia model also captures some interesting aspects of the movement selection rate that the other models do not capture. The children's data show that, when starting at distance 5 inner ring nodes, the first move that is selected is wrong more often than if the children moved randomly (Fig. 4.b). A similar behavior is observed in $\langle H{=}2, nr \neq 0\rangle$, although for a different set of distance 5 states (outer ring nodes instead of inner ring nodes). This behavior is not observed in the non-inertial models, which do not become sufficiently "stubborn" to perform below chance in certain specific initial configurations, and always show a first move selection rate that is biased towards the correct move.

This is most likely due to the fact that for non-inertial or Markov agents, the selection rate of actions in a given state is independent of the previously visited states. Thus, if the model biases the movement selection rate in a state at distance 5 from the goal toward the wrong move, the distance 6 and 7 nodes, whose correct game play must pass through this state, would be greatly affected. As the children play very well in the antipode, the model must balance the movement selection bias in order to partly fit the good gameplay in the antipode and the bad gameplay at distance 5 nodes. However, the non-Markov agents can produce this bias without greatly affecting the performance at distance 6 or 7 because of the inertial movement selection rate. The persistence in systematic errors is a stubbornness of model-based agents: they "trust" a model which works well on average but fails in certain specific configurations (Sutton & Barto, 1998).

These results were based on an analysis of performance from any given configuration. The differences in behavior between inertial and non-inertial models are expected to be more pronounced in intermediate actions throughout the trajectory of a trial. Children's performance, like the inertial model's performance, is very high (above 80%) for all distances (Fig. 3.c). In other words, if the first move is correct, then children have a very high chance of making it to the goal because they persist in the same path. This feature is not expressed by the non-inertial models, whose performance does not change very much as a function of the first move's correctness.

These observations support the working hypothesis of non-Markov behavior (B in Sec. 1.2).

Although the $\langle H{=}2, nr \neq 0\rangle$ model can describe many aspects of the data, it cannot account for the entire set of observations. For instance, it cannot explain the non-symmetric movement selection rate for states that are to the left or to the right of the goal and antipode in the game-state graph (Fig. 4). As mentioned above, our intention here was not to achieve a perfect fit with a very complicated model which would describe highly domain specific aspects of the data.

A concern about our model is that it assumes that children have partial access to the correct representation of the game K, which seems unlikely and, moreover, we cannot offer precise ways as to how this representation can be computed. The logic for this simplification is as follows. First, greedy distance estimates and inertia cannot be the only inputs to the value function available to children. If this were the case, the first move selection rate should always be biased toward the local minimums of greedy distance. This is not observed in distance 6 and distance 5 outer ring states (Fig. 4), which were close enough to the antipode to be able to suffer such bias. Also, many states form a sort of greedy plateau in the graph (distance 4, 5 and 6 states). To traverse these states, children require resources beyond the greedy heuristic. We reasoned that inertia may serve this purpose and modeled an agent dictated only by the herustics of inertia and greediness ($nr \neq 0$ and $\eta = 0$). Results show that the model's performance is significantly worse than the children's performance ($p < 10^{-18}$ $df = 1$), but also showed increased performance in the antipode (Fig. 6). The latter observation is a consequence of inertia and indicates that the greedy value plateau is crossed with this extra resource. However, the fact that children's performance is much higher than the model's points to the fact that children have access to other resources to generate a value function beyond greedy and inertia heuristics.

A second concern, is that the value function is completely deterministic. To solve this problem we ran a model with an additional parameter which adds random noise to the value function (eq. 2). For high values of σ the value function is simply random. For low *sigma* the value function represents partial knowledge, corrupted by a heuristic of greediness. Results (Fig. 6) show that there is a strong concave zone for the mean squared difference in the pair $\{\sigma, \beta\}$ which reaches a minimum for $\sigma \sim 0$ and β values similar to the fits of the non-extended model.

Hence, the results of simulations with the additional parameter σ, which adds noise to the value function, show quite conclusively that the best fit is obtained for an almost noiseless value function $\sigma \sim 0$. Moreover, for the lower values of σ, there is a broad range of β values that fit the data reasonably well, indicating that the injection of non-zero noise in the value function results in an agent which is less robust to noise, or has an imperfect determination of the search parameter β. This may seem counter-intuitive, since one could assume that higher values of σ may in fact reflect a better adjustment of an imprecise value function (there is no rationale to assume that children have a perfect representation of the

game, but instead several resources which may approximate it). This shows that the attempt to adjust this "intermediate knowledge" as a noisy version of perfect knowledge turned out not to work, implying that children must have a list of available resources that approximate perfect performance of the game in a deterministic way (better than simply inertial and greedy behavior which, as we showed above, cannot account for the data).

Another simple computational account of how partial knowledge of the game may be incorporated into the value function is that children may know that the game is solvable from certain states. This is typical in many planning games where states can be divided in two classes: 1) a subset of states which are "known territories", where value can be calculated directly or can be looked up in a table, and 2) a more complex domain (typically, further from the goal) where the only resources are heuristics. As an analogy, a chess player may feed the value function distinctively from heuristics or from game knowledge in different states. In complex positions the value function has to be based on heuristics (number of pieces, number of threats, mobility of pieces, occupation of key squares...). Instead the player may know that certain positions - for instance a rook and king vs king ending - are won, and have a precise, known and deterministic procedure to solve them. In our model, this is equivalent to saying that η depends on the state. For certain states children may have full knowledge of the game (i.e. $\eta = 1$) while for other states (presumably at greater distances) they might only rely on heuristics.

Equivalently, this scenario can be modeled with a function K that encodes the distance to the goal in a subset of states and assumes a uniform value for all other states. Hence, the minimization of the value function is determined by K in this subset of known states and by the heuristic in the remaining states where K is homogeneous, revealing lack of knowledge. In the simplest case, the subset of states is determined by a horizon of a given X (i.e. all the states with $d < X$). [4]

If all children have the same horizon of known states, our data could be described by a value function based on $K+Heu$ (Heu for heuristic), where K assumes low and precise values for the subset of known states ($d < X$) and dominates the value function. The contribution of Heu becomes relevant only in the unknown states in which K saturates.

Analysis shows that the data is inconsistent with the hypothesis of a fixed horizon of perfectly known states X: all functions in which K saturated at a horizon showed consistently worse fits (see Supplementary Fig. B).

3.4 Modeling Children's variability

In this section we zoom in on $\langle H{=}2, nr \neq 0 \rangle$, analyzing the mean performance at each starting distance, assuming that each child's data are independent from one another. Performance varies widely between the children (Fig. 7). The broadest variability is observed, as expected, for the more difficult conditions of $D = 5$ and 6. A raster plot shows that changes in performance of the model, as with children, are mainly governed by problems with starting configurations at these distances. We perform a linear regression between children and their corresponding model performance for $D = 5$ and $D = 6$ separately. This shows a significant correlation (for $D = 5$, standard coefficient 0.64, $p = 2.10^{-8}$, $df = 71$ and for $D = 6$, standard coefficient 0.38, $p = 6.10^{-5}$, $df = 71$). Moreover, the model does not show significant differences in the variance in these conditions. However, for the shorter distances, the model shows a greater dispersion (Fig. 7).

We test whether the model has the same variance in performance as the children at each starting distance with a Levene test (Levene, 1960) (the null hypothesis is that the variances are the same). This analysis shows that the model's variability does not significantly differ for the longer distances, but does differ for $D = 2$ and $D = 3$. For tests with Bonferroni and Holm - Bonferroni corrections, $D = 2$ is the only distance at which the variances significantly differ (table 2).

All children, even those that have the lowest mean performance, are virtually perfect in these trials, and the model shows more variance and worse performance. The model is not perfect for $H = 2$ at distance 2 because it generates plans stochastically (with a distribution determined by β), and some of these plans never reach the goal. A natural extension of this model which could solve this problem would have β be dependent on the distance. In practical terms, this may reflect the fact that children do not play in the same way for simple ($D = 2, 3$) and hard ($D = 5, 6$) problems, much in the same way that mathematical addition of small numbers uses different cognitive resources (memory, verbal operations...) than addition of large numbers (Dehaene, 1997).

An important aim of this investigation is to determine whether individual parameters obtained from the model are informative about children's performance beyond this specific game setting. Our hypothesis (C), described in Sec. 1.2), is that parameters which affect the value function should show little transfer, while search parameters should instead transfer to novel problems. To examine this hypothesis, we measured each child's parameters: β, γ, η and nr (see Table 1) where η, nr and γ conform the value function (see equation 1). For $\langle H{=}2, nr \neq 0 \rangle$, $\beta = 9.5 \pm 4.8$, $\gamma = 0.46 \pm 0.30$, $\eta = 0.44 \pm 0.21$ and $nr = 0.50 \pm 0.26$.

We first investigated the correlation between the model's parameters and each child's mean performance. Mean performance covaries positively with η (0.18 ± 0.13). This is expected, since it simply states that children whose value function relies more on game knowledge and less on a greediness heuristic have a greater chance of choosing correct moves.

[4] This must not be confused with the planning horizon.

In contrast, β, γ and nr do not show a significant correlation with performance (0.01 ± 0.12, -0.09 ± 0.10 and 0.11 ± 0.13 respectively). These parameters also show expected interactions. For example, it is reasonable to expect that the group of children that tend to select their moves more randomly (low β obtained from fit) will not show a correlation between η and performance. We confirm this by comparing the cross correlation between the mean performance and η for the children that have the 25% lowest β and the 25% with the highest β. The correlation for low (high) β is equal to 0.05 (0.57), with the p-value of being randomly correlated equal to 0.41 (0.008) (Fig. 8).

This analysis is largely expected and merely confirmatory. The most interesting question is if the obtained model parameters are indicative of the children's performance in a different game. To this aim, we correlated parameters of the model with performance in the Tower of London (ToL) game, measured in two independent sessions, several days before the beginning and after the completion of the DCM game. To measure the correlation between the model's parameters and DCM and ToL performance, we split the children's data into five groups (their corresponding school classes). We then fit a linear regression between the model's parameters and the performance measures. In DCM, the performance measure is the fraction of times the children arrived at the goal state in the minimum number of moves. In ToL, the model parameters were regressed against many performance markers (if the goal had been reached, if it was done in the minimum amount of moves, and if the maximum difficulty was reached). We then perform a t-test over the regressors and use the standard coefficients to measure the strength of the correlations. The null hypothesis was that the coefficient's mean value was equal to 0 and thus the model parameter was not indicative of ToL performance. We observe that η, which had a major influence in DCM, does not show significant transfer to performance in ToL ($p = 0.31$, $df = 29$). Instead, the discount rate (γ), and most notably the stochasticity of branch selection in the search process (β), show significant transfer indicative of children's performance in ToL ($p = 0.03$ and $df = 29$ for both). Parameter nr does not show significant transfer either ($p = 0.1$ and $df = 29$).

We also performed the test adding DCM total mean performance as a regressor. We find that this is significantly correlated to ToL performance, but the parameter β remains significant ($p = 0.03$ $df = 29$). However, the parameter γ ceases to be significant ($p = 0.11$). These results are in line with our working hypothesis that the non-game-specific parameters (β) should show transfer to other planning tasks, while the game-specific parameter (η, nr and γ[5]) should not (C in Sec. 1.2).

4 Discussion

Our work is inspired by Klahr's (Klahr, 1985) in that we 1) used the Dog-Cat-Mouse game to investigate children's performance in a friendly version of an N-puzzle 2) modeled the children as stochastic agents that "walk" through the game state graph and 3) aimed to detect "weak methods" and search strategies used by children to solve the game. The main differences from Klahr's original work (Klahr, 1985) are 1) the investigation of they dynamics of play and 2) generating a large corpus of data with several sessions of play per child, which allowed us to parse out different aspects of the planning strategy (Geffner, 2010): heuristics and knowledge of the game, which contribute to a value function, and search strategies to define actions based on the distribution of a value function. The models presented here, which use more sophisticated mathematical constructions and are derived from a large corpus of data, satisfied the expectation of performing better than Klahr's agents (see Supplementary C). In fact, only a few distances and states were sampled in Klahr's original paper, and the observation was that distance was a bad predictor of performance. Here we zoomed in on this result, revealing a very idiosyncratic pattern of dependence on distance that allows us to identify children's computational resources for problem solving. Using this approach, we were able to separate the contributions made by how randomly children selected their moves, and the mechanism that guided their selection.

Our assumption, and the core of the model presented here, is that action selection is stochastically guided by the combination of a slow recursive simulation of possible outcomes (the search process) and the use of a fast value estimation (that uses both knowledge and heuristic functions). Direct behavioral observations revealed several principles which justify the choice of this class of models.

First, the data revealed mostly stationary levels of performance (Fig. 2). Children played well above chance from the very first trial. This demonstrates that they do not need to go through a slow trial and error learning process to acquire resources that allow them to play the game.

Second, the data showed stubbornness and error persistence in certain specific game configurations (Fig. 4.a). These are typical features of models - like the one we proposed - based on stochastic search of value functions. To achieve accurate performance (as children do), the model has to have relatively low levels of stochasticity in the search process (high β), which is not sufficient to prevent the planner from falling repeatedly at local minima of the value function.

Third, the movement selection rate significantly differed in several states depending on the history of previous moves

[5] Although nr and γ are not game specific, they are part of the value estimation module and are strongly correlated to the game-specific heuristic and knowledge

(Fig. 5). This indicates that performance cannot be described by a Markov agent that makes decisions based only on the present game configuration without taking into account how that state was reached. This justifies the inclusion of a no return cost in the value function. The incorporation of such a procedure in the value function is reminiscent of the sunk cost in behavioral economics (Arkes & Blumer, 1985), which refers to a past cost that has already been incurred and cannot be recovered, but which nevertheless conditions subsequent actions leading to non-Markov behavior.

Our modeling approach can be linked to the field of bounded rationality (Gigerenzer & Selten, 2002) which states that perfectly rational decisions are often not feasible in practice, because of the finite computational resources available for making them. The rationality of individuals is limited by certain facts, such as the information available, the amount of time to make a decision and cognitive abilities. The "weak methods" that children are observed to use may be viewed from this standpoint. However, we believe that within this general class of ideas, our model adds some specific concepts. Above and beyond their lack of sufficient knowledge, children have specific stereotyped gameplay tendencies such as avoiding back-up moves. Inertia may be an adequate resource in certain circumstances, but it prevents perfect play from a general standpoint.

This set of observed behaviors justify the choice of the model's class. This justification is important because our fitting procedure can select which of a number of models explains the data better, but cannot inform how candidate models are generated in the first place. After choosing the class of models (stochastic planners), fitting serves two purposes. First, to infer the parameters such as depth search, which are meaningful variables of children's thought. Second, to ask whether these parameters are indicative of a repertoire of a child's general resources, which may serve to predict performance in different tasks.

Individual performance in two structurally identical problems may differ due to domain specificity. Our modeling effort can be seen as a way to factor out domain specific procedures which contribute to the value function. In agreement with this expectation, we show that search specific parameters of the model are the ones that are informative for predicting performance in a different planning task (Sec. 3.4).

Our work has several limitations which should be solved in future research. First, during our work we assume a given form of the heuristic function and game knowledge. We expect that these are not the true estimation methods, but general approximations that are suitable for the DCM game. We certainly acknowledge, that the inclusion of a correct distance knowledge as a weighted factor in the value function is a simplification of the problem solving resources. Second, a very important the puzzle which should be resolved by future work is how to incorporate domain independent and scalable heuristics, that are constructed using general procedures such as, for instance, the relaxation method (Hoffmann & Nebel, 2001; Keyder & Geffner, 2008; Pearl, 1984), into the model instead of using heuristics that are encoded by hand.

Acknowledgements

This research was supported by Consejo Nacional de Investigaciones Científicas y Técnicas, Ministry of Science of Argentina, Human Frontiers, and Fundación Conectar.

We wish to thank Audrey K. Kittredge for very carefully proofreading the manuscript and for providing numerous useful and very constructive suggestions.

We also wish to thank Varinia Telleria for the design of characters and drawings, and Sebastian Lipina, Julia Hermida, Soledad Segretin, Martin Elias Costa, Matias Lopez-Rosenfeld and Diego Fernandez Slezak for helping and building Mate Marote.

M.S. is sponsored by the James McDonnell Foundation 21st Century Science Initiative in Understanding Human Cognition—Scholar Award.

References

Arkes, H. R., & Blumer, C. (1985). The psychology of sunk cost. *Organizational behavior and human decision processes*, *35*(1), 124–140.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., ... Colton, S. (2012). A survey of monte carlo tree search methods. *Computational Intelligence and AI in Games, IEEE Transactions on*, *4*(1), 1–43.

Dehaene, S. (1997). *The number sense: How the mind creates mathematics*. Oxford University Press.

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association*, *56*(293), 52–64.

Geffner, H. (2010). Heuristics , Planning and Cognition. In R. Dechter, H. Geffner, & J. Y. Halpern (Eds.), *Heuristics, probability and causality. a tribute to judea pearl* (pp. 23–42). College Publication, UCLA.

Gigerenzer, G., & Selten, R. (2002). *Bounded rationality*. Cambridge: MIT Press.

Goldin, A. P., Hermida, M. J., Shalom, D. E., Costa, M. E., Lopez-Rosenfeld, M., Segretin, M. S., ... Sigman, M. (2014). Far transfer to language and math of a short software-based gaming intervention. *PNAS*, *in press*.

Goldin, A. P., Segretin, M. S., Hermida, M. J., Paz, L., Lipina, S. J., & Sigman, M. (2013). Training Planning and Working Memory in Third Graders. *Mind, Brain, and Education*, *7*(2), 136-146. doi: 10.1016/j.neuron.2011.02.027

Hoffmann, J., & Nebel, B. (2001). The FF Planning System : Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research*, *14*, 253–302.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, 65–70.

Keyder, E., & Geffner, H. (2008). Heuristics for Planning with Action Costs Revisited. In *Proceedings of the 18th inter-*

national conference on artificial intelligence (pp. 588–592). Amsterdam, Netherlands.

Klahr, D. (1985). Solving Problems with Ambiguous Subgoal Ordering: Preschoolers' Performance. *Child Development*, *56*, 940–952. doi: 0009-3920/85/5604-0020\$0100

Klahr, D., & Robinson, M. (1981). Formal Assessment of Problem-Solving and Planning Processes in Preschool Children. *Cognitive Psychology*, *13*, 113–148.

Kocsis, L., & Szepesvári, C. (2006). Bandit Based Monte-Carlo Planning. In J. Furnkranz, T. Scheffer, & M. Spiliopoulou (Eds.), *Machine learning: Ecml 2006 lecture notes in computer science* (Vol. 4212/2006, pp. 282–293). Springer Berlin / Heidelberg. doi: 10.1007/11871842_29

Levene, H. (1960). Robust tests for equality of variances. In I. Olkin, S. G. Ghurye, W. Hoeffding, W. G. Madow, & H. B. Mann (Eds.), *Contributions to probability and statistics: essays in honor of harold hotelling* (p. 278-292). Stanford University Press.

Lopez-Rosenfeld, M., Goldin, A. P., Lipina, S., Sigman, M., & Slezak, D. F. (2013). Mate marote: a flexible automated framework for large-scale educational interventions. *Computers & Education*(0), - . Retrieved from `http://www.sciencedirect.com/science/article/pii/S0360131513001462` doi: 10 .1016/j.compedu.2013.05.018

McCormack, T., & Atance, C. M. (2011, March). Planning in young children: A review and synthesis. *Developmental Review*, *31*(1), 1–31. Retrieved from `http://linkinghub.elsevier.com/retrieve/pii/S0273229711000049` doi: 10.1016/j.dr.2011.02.002

McDermott, D. (1996). A Heuristic Estimator for Means-Ends Analysis in Planning. In B. Drabble (Ed.), *Proceedings of the 3rd international conference on artificial intelligence planning systems (aips-96)* (pp. 150–157).

Newell, A., Shaw, J. C., & Simon, H. (1959). Report on a general problem-solving program. In *Proceedings of the international conference on information processing* (pp. 256–264).

Pearl, J. (1984). *Heuristics: Intelligent search strategies for computer problem solving*. Addison-Wesley.

Péret, L., & Garcia, F. (2004). On-line search for solving Markov decision processes via heuristic sampling. In R. López de Mantarás & L. Saitta (Eds.), *Proceedings of the 16th european conference on artificial intelligence (ecai)* (pp. 530–534). Amsterdam: IOS Press. Retrieved from `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.4531&rep=rep1&type=pdf`

Piaget, J. (1976). *The grasp of consciousness: Action and concept in the young child*. Cambridge, MA: Harvard University Press.

Plackett, R. L. (1983). Karl pearson and the chi-squared test. *International Statistical Review*, *51*(1), 59–72. (doi:10.2307/1402731)

Puterman, M. L. (1994). *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons.

Roesch, M. R., Calu, D. J., & Schoenbaum, G. (2007). Dopamine neurons encode the better option in rats deciding between differently delayed or sized rewards. *Nature Neuroscience*, *10*(12), 1615–1624.

Russell, S. J., & Norvig, P. (1995). *Artificial intelligence: a modern approach*.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press.

Variable	Meaning
β	The Softmax inverse temperature that governs the softmax search probability distribution (eq. 3). A high value of β yields a greedy search that only expands the highest valued states whereas low β gives purely random state selection.
γ	The reward discount γ maps the distance estimate $D(s_{t+1})$ to a value estimate. It's an implementation of a delayed reward exponential decay analogous to the reward discount in the RL and MDP frameworks (Sutton & Barto, 1998).
η	The mixture factor between partial knowledge K and heuristic G distance values for a given state s_{t+1} (Eq. 1).
nr	The value penalization of a backup move. It's used in the $nr(s_{t+1}, s_{t-1})$ non-Markov function (eq. 1) in order to penalize the value of the state s_{t+1} if it is the same as s_{t-1}.
σ	The standard deviation of the gaussian noise added to the state value estimate (eq. 2). This noise can factor out the weight of greediness and the quality of the internal representation of the game.
H	The plan horizon or the maximum search depth.
R	Number of rollouts. It refers to the number of plans searched before selecting the highest end valued state.

Table 1
A detailed list of the planning model variables and their meaning.

Distance	D = 2	D = 3	D = 4	D = 5	D = 6	D = 7
p-value	0.006	0.019	0.52	0.63	0.12	0.24

Table 2

The Columns contain the data's Levene test p-value for each starting distance. A Bonferroni and a Holm - Bonferroni (Holm, 1979) corrected test yields only significant differences for $D = 2$.

Figure Captions

Fig. 1: The 4 upper panels show a game play example. Below these, the graph of the game states is plotted. The color circles represent each character (pink = girl, blue = boy, green = cat). The squares stand for the tree-houses and the colors indicate to which character it belongs. The red arrow represents the game trajectory of the upper panels in the graph. The states are labeled by the numbers next to them.

Fig. 2: The upper panel shows the mean performance over all the children that reached each trial. The curve is smoothed with a running average of 10 trials and the darker color zone shows the standard deviation of the mean. The horizontal slashed lines are the mean performance over all trials. The lower panel shows the performance of each child in every trial. It shows that performance drops are observed in almost all children, which cannot be seen from the upper panel. This raster also shows that there is a density of mistakes in the majority of children sustained throughout the experiment. The subjects were arranged to have a growing amount of trials in the vertical axis. The black patches in this graph indicate that the trial was won in the minimum amount of moves and the gray ones indicate it wasn't. The white patches correspond to having no data for the corresponding trial. During phases 1 and 2 of the game, children face a sequence of trials that have increasingly distant states. They start from $D = 2$ and go up to $D = 7$. As this progression is ordered, during the first trials most children play at the same starting distance. The horizontal colorscheme indicates which starting distance was the most common for the given trial played by the children. It spans up to the trial where more than half the children were playing in phase 3. The vertical dashed lines indicate transitions to very difficult starting distance states that are accompanied by a large drop in performance.

Fig. 3: a) contains the mean performance curve over the starting distances. The error bars correspond to the standard deviation of the mean performance for each starting distance. b) The two middle panel line plots are the children's and fitted models performance vs distance. The left panel correspond to the Markov (without inertia) models fit. The right panel shows the non-Markov model fit. c) shows the mean performance vs distance of the children and planning models with $H = 2$ with and without inertia. The darker colored lines are the performance curves after a first correct move and the lighter toned, show the total mean performance. The N in the legend is the number of correct first moves done by the children or the model's simulations.

Fig. 4: a) shows in a blue scale, the children's mean performance for each state and in red, the first move selection rate. Each graph node corresponds to the states drawn in figure 1. The green node indicates the position of the goal and the transparent nodes are states from which the game never started. b) The panels show the fitted models mean performance in the state-space graph with their corresponding first move selection rate. The color-scales of the mean performance and of the first move selection rate are the same for all plots.

Fig. 5: Movement selection rate for the entire children dataset. The bars are grouped in the horizontal axis according to the states at which they took place and are labeled with the state numbers as appear in Fig. 1. The bars at a given state correspond to game moves that could take place in it. The green bars are assigned to correct moves in the state and the red bars to wrong ones. The horizontal axis also shows the distance from each state to the goal. The upper panel shows the first move selection rate and the lower one shows the move selection rate when passing through a state if the first move was correct (PAFC, Passing After First Correct). There are visible differences between the first and PAFC move selection in the states at distances 4 through 6, that indicate a non-Markov movement selection policy.

Fig. 6: a) Shows the children's total mean performance vs distance in black. The color lines are the model fits for $H = 1$, 2 and 3, with $\eta = 0$. b) and c) use mean squared difference χ^2 that is obtained from fits of the extended planning model. b) shows the mean over all children χ^2 divided by their largest χ^2 value. The horizontal axis corresponds to the values of β that in the left panel go from 0 to β_{opt} (that was obtained fitting the model without sigma) and in the right panel from β_{opt} to ∞. The vertical axis corresponds to the values of σ and go from 0 to σ_{opt}. The latter is the value obtained after fitting the model with $\beta \to \infty$. If $\sigma_{opt} < 0.01$, we used $\sigma_{opt} = 0.01$. The parameter values used as σ_{opt} and β_{opt} were distributed as $\sigma_{opt} = 0.04^{+0.16}_{-0.03}$ and $\beta_{opt} = 9.5 \pm 4.8$. c) shows the mean over the values of σ of the data plotted in (b) in a logarithmic scale.

Fig. 7: a) Each dot corresponds to a child performance vs it's fitted model performance. Each color is associated to a certain initial distance labeled in the legend. Distance 5 and 6 plots (with a fitted linear regression in dashed lines) are shown separately at the right side of the legend. b) Shows the mean performance for each distance and each subject in a gray scale (white is mean 0 and black is performance 1). The subjects index (vertical axis) for both panels are arranged to have a growing total mean performance for the children. The higher panel corresponds the children data and the lower corresponds to the corresponding fitted model. c) plots the difference between each child's and corresponding fitted model's mean performance. Each panel shows the data for a given starting distance state. The subject index was sorted as to have a growing mean total performance.

Fig. 8: Children's total mean performance vs the fitted η value. The size of the markers is proportional to the number of trials played by the child and the color corresponds to the

fitted β value. The color-scale is shown in the colorbar next to the scatterplot.

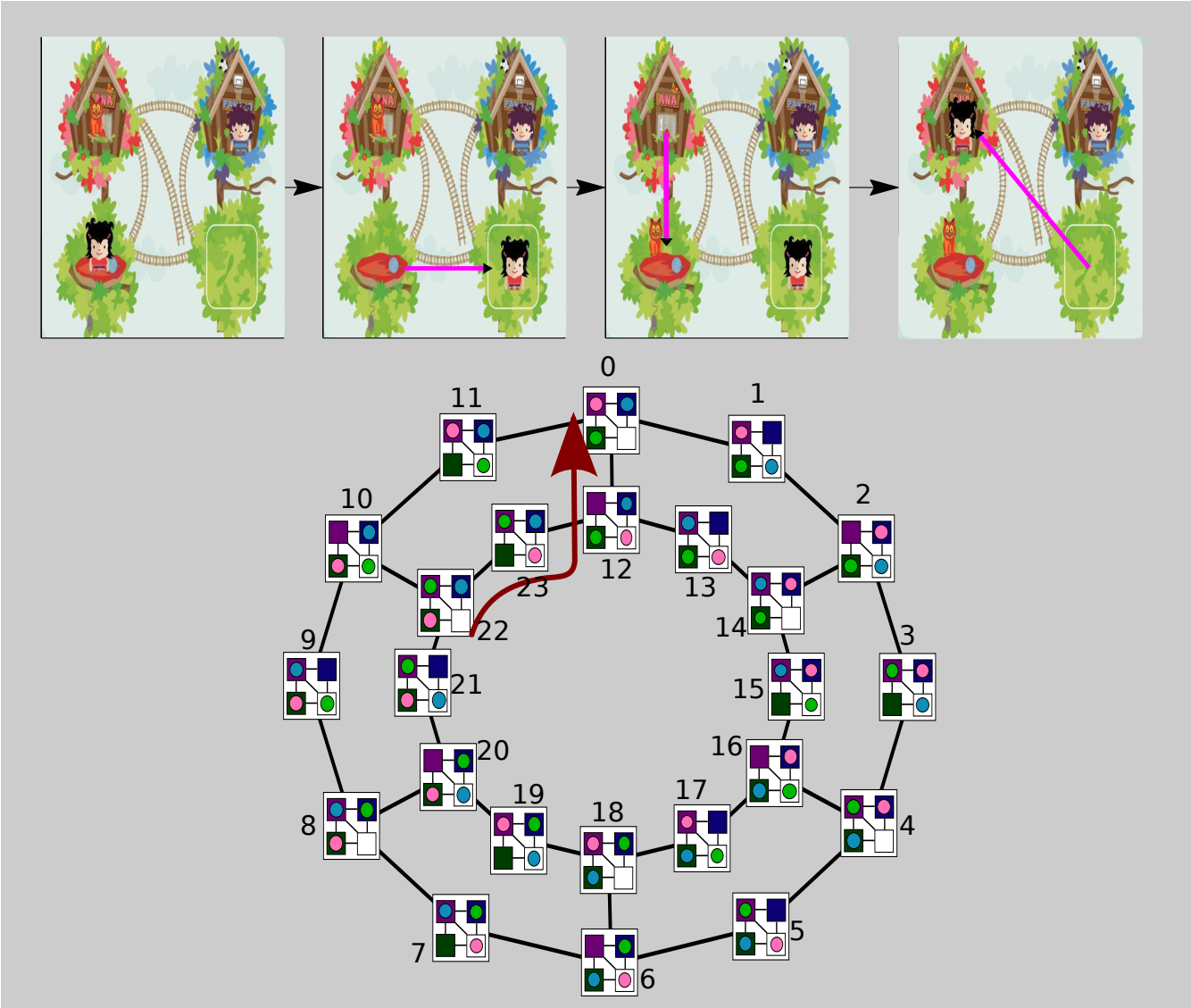

Figure 1

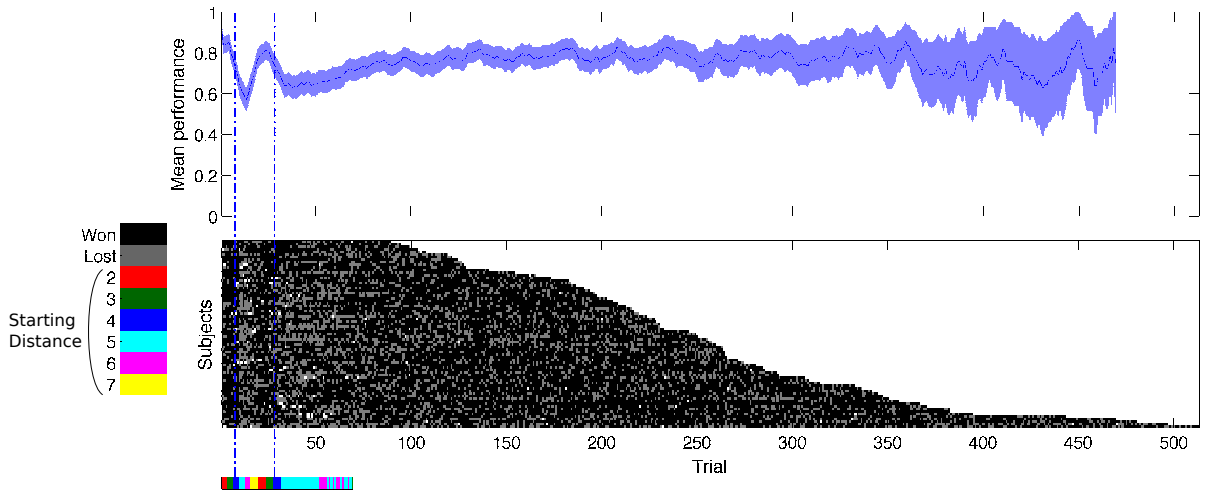
Mean performance
1
0.8
0.6
0.4
0.2
0
Won
Lost
Starting Distance
2
3
4
5
6
7
Subjects
50
100
150
200
250
300
350
400
450
500
Trial

Figure 2

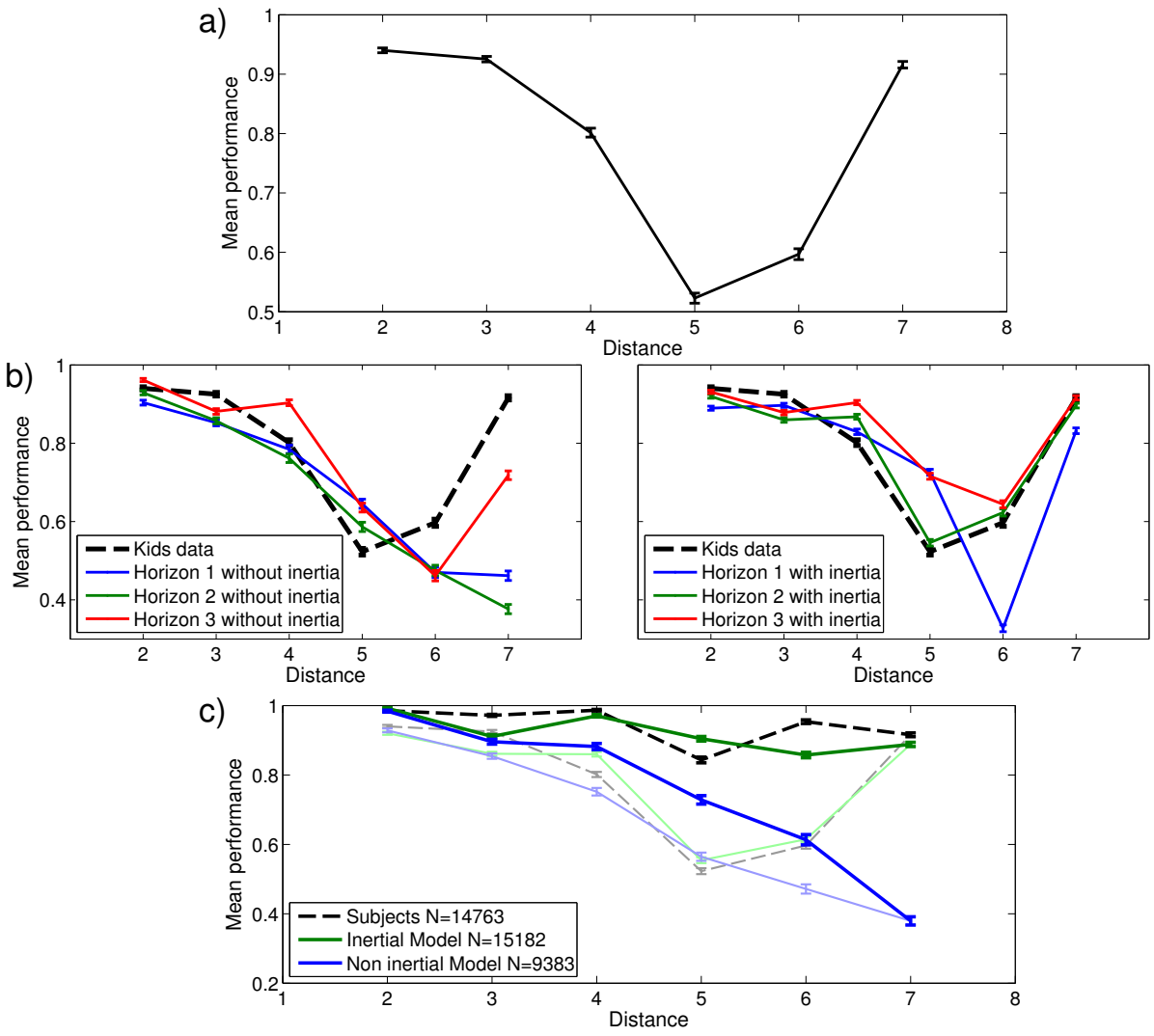
a)
Mean performance
Distance
b)
Mean performance
Kids data
Horizon 1 without inertia
Horizon 2 without inertia
Horizon 3 without inertia
Distance
Kids data
Horizon 1 with inertia
Horizon 2 with inertia
Horizon 3 with inertia
Distance
c)
Mean performance
Subjects N=14763
Inertial Model N=15182
Non inertial Model N=9383
Distance

Figure 3

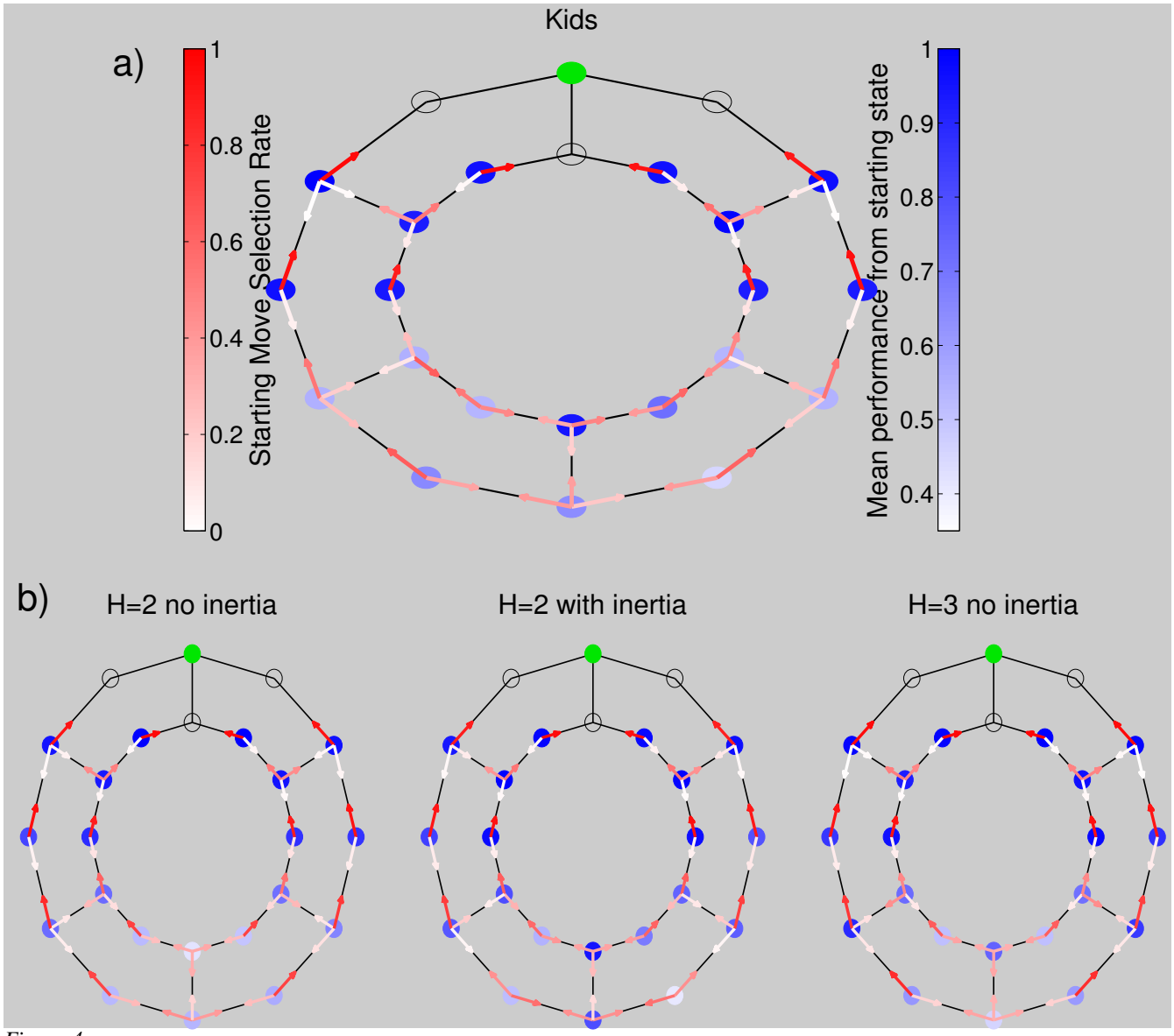
Kids
a)
Starting Move Selection Rate
1
0.8
0.6
0.4
0.2
0
Mean performance from starting state
1
0.9
0.8
0.7
0.6
0.5
0.4
b)
H=2 no inertia
H=2 with inertia
H=3 no inertia

Figure 4

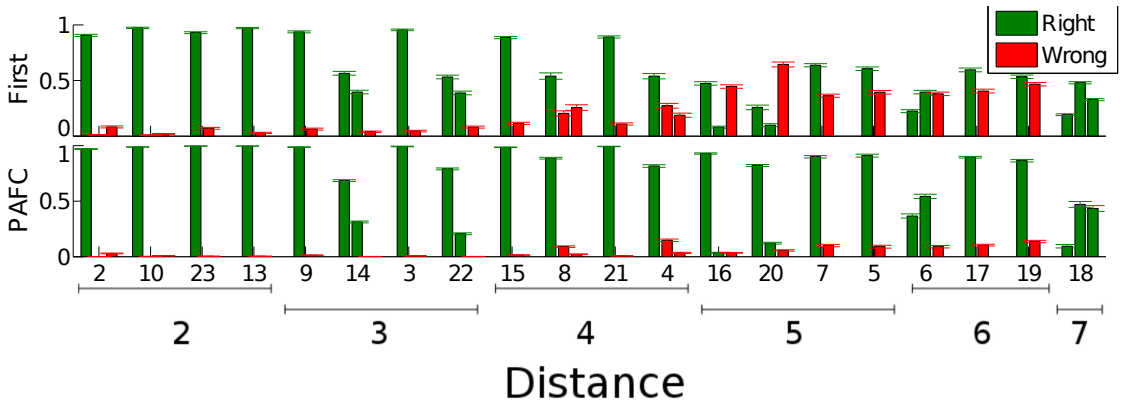
Right
Wrong
First
PAFC
1
0.5
0
2 10 23 13 9 14 3 22 15 8 21 4 16 20 7 5 6 17 19 18
2
3
4
5
6
7
Distance

Figure 5

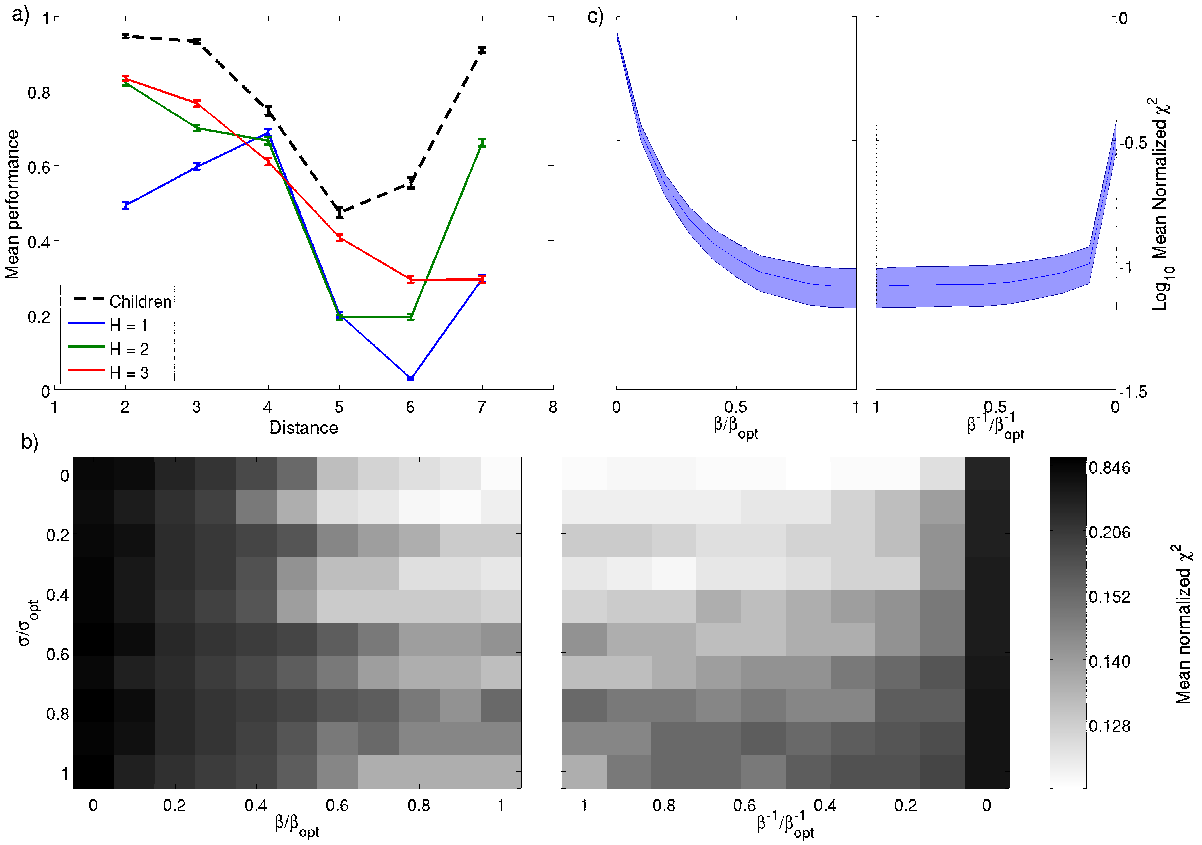
a)
Mean performance
Distance
Children
H = 1
H = 2
H = 3
c)
β/β_opt
β^-1/β^-1_opt
Log10 Mean Normalized χ2
b)
σ/σ_opt
β/β_opt
β^-1/β^-1_opt
Mean normalized χ2
0.846
0.206
0.152
0.140
0.128

Figure 6

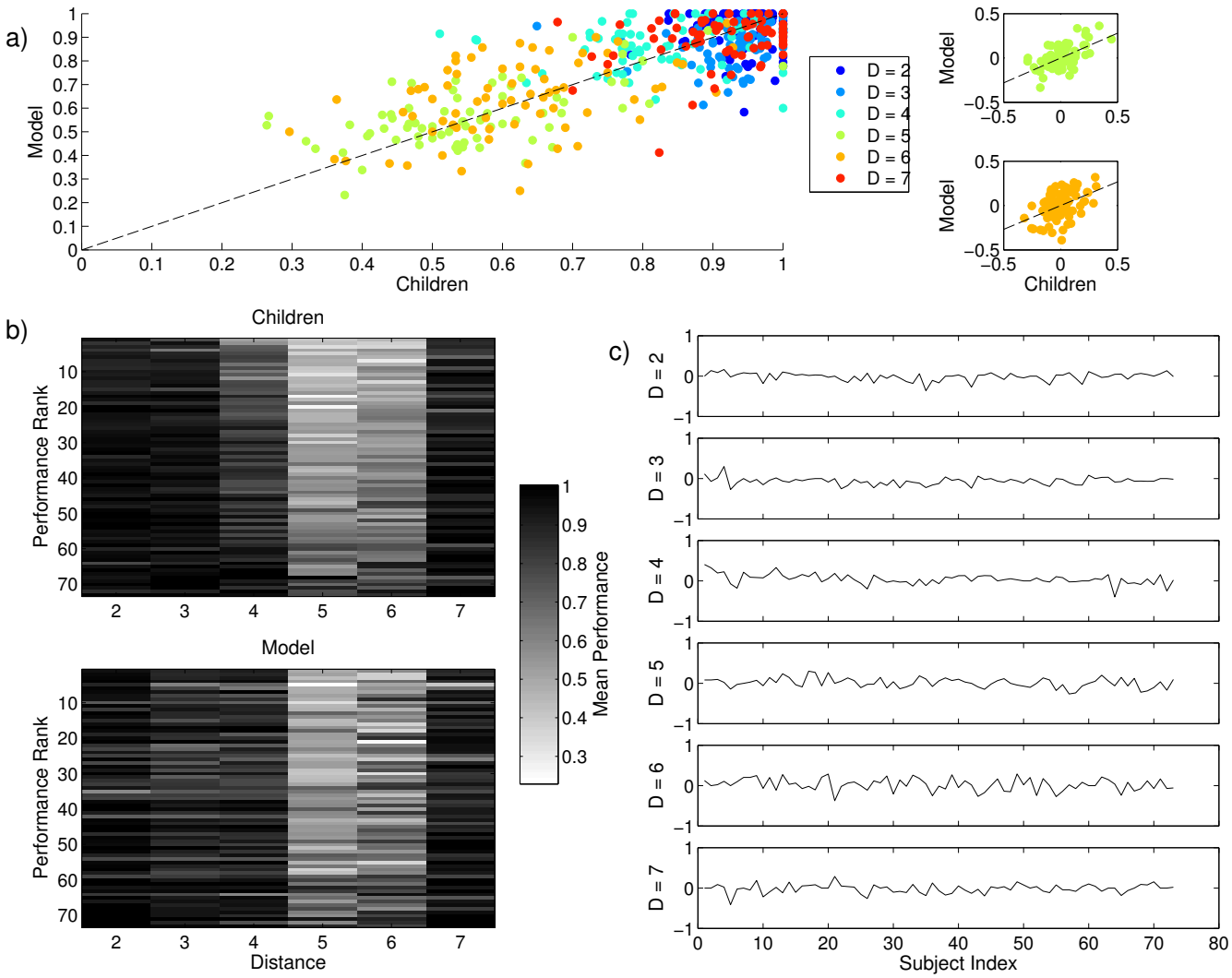
a)
Model
Children
D = 2
D = 3
D = 4
D = 5
D = 6
D = 7
b)
Children
Performance Rank
Model
Distance
Mean Performance
c)
D = 2
D = 3
D = 4
D = 5
D = 6
D = 7
Subject Index

Figure 7

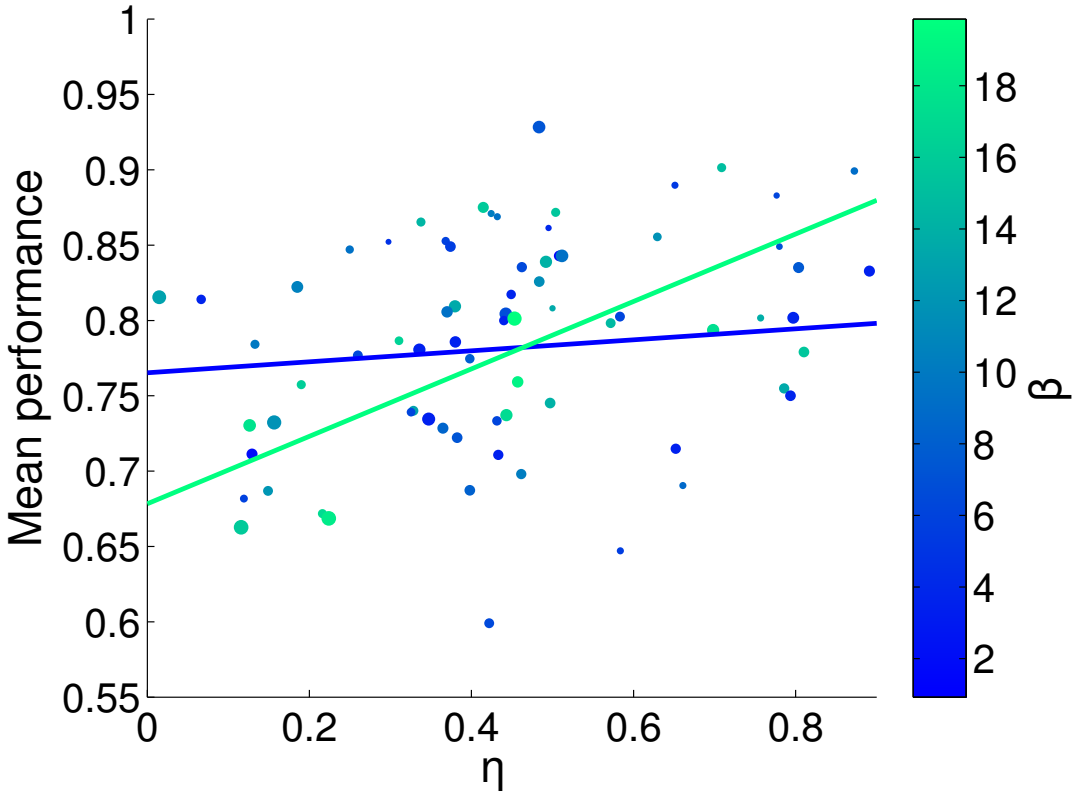
Mean performance
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0
0.2
0.4
0.6
0.8
η
18
16
14
12
10
8
6
4
2
β

Figure 8