# Complexity and Procedural Choice[*]

James Banovetz[†] Ryan Oprea[‡]

January, 2022

**Abstract**

We test the core ideas of the "automata" approach to bounded rationality, using simple experimental bandit tasks. Optimality requires subjects to use a moderately complex decision procedure, but most subjects in our baseline condition instead use simpler (often suboptimal) procedures that economize on "states" in the algorithmic structure of the rule. When we artificially remove the mental costs of tracking states by having the computer track and organize past events, subjects abandon these simpler rules and use maximally complex optimal rules instead. The results thus suggest that the main type of complexity described in the automata literature fundamentally influences behavior.

**Keywords:** Complexity, automata, procedural decision making, bounded rationality, bandit problems, economics experiments

**JEL codes: C91, D91 G0,**

[*] We thank Douglas Bernheim, Alex Imas, Muriel Niederle, Emanuel Vespa and Sevgi Yuksel for their valuable comments and feedback. This research was supported by the National Science Foundation under Grant SES-1949366 and was approved by UC Santa Barbara IRB.

[†] Banovetz: The Brattle Group, One Beacon Street, Suite 2600, Boston MA 02108, https://www.orbitz.com/trips james.banovetz@gmail.com;

[‡] Oprea (corresponding author): Economics Department, University of California, Santa Barbara, Santa Barbara, CA, 93106, roprea@gmail.com.

# 1 Introduction

There is a long tradition in economics arguing that humans (i) dislike using complex rules to make decisions (Simon 1955), and (ii) economize on this "procedural complexity" by substituting to simpler (often suboptimal) rules instead. In the last few decades, the "automata literature" in economic theory (e.g. Aumann 1981, Neyman 1985, Rubinstein 1986, Abreu & Rubinstein 1988, Kalai & Stanford 1988) has formalized this idea by modeling decision procedures (e.g. strategies in games) as *finite state machines*, models of algorithms developed in computer science. By advancing hypotheses about what makes one procedure/algorithm more "complex" (and therefore costly) to implement than another, this literature provides a theoretical framework for studying the influence of complexity on human behavior. The original aim of this literature was to ease the indeterminacy of the folk theorem by using complexity to refine the set of feasible strategies in games and thus produce more predictive theory. In the decades since, this approach to bounded rationality has proven equally successful at rationalizing phenomena ranging from the emergence of Walrasian outcomes in markets (e.g. Sabourian 2004, Gale & Sabourian 2005) to some of the key anomalies documented in behavioral economics (e.g. Salant 2011, Kalai & Solan 2003, Chauvin 2020, Wilson 2014).

In this paper we provide a crisp experimental test of the core ideas from this literature. Specifically, we evaluate the hypothesis that behavior in a canonical task is fundamentally shaped by deliberate efforts to avoid the specific type of complexity ("state complexity") that motivates the automata literature. Instead of studying repeated games (which introduce a number of complications orthogonal to our question), we study the closest analogue individual decision making task: a two-arm "bandit" problem. In this task, subjects repeatedly choose between two "arms," $a_1$ or $a_2$, each of which offers a fixed per-period payment. The first arm the subject selects (regardless of which arm she selects, $a_1$ or $a_2$) is guaranteed to always pay a known intermediate value $x$ (in our implementation $x = 0.65$). The other arm (the arm that the subject did *not* initially choose) pays 0, $x$ or 1 with equal ex ante likelihood. Under our parameters, subjects should always "explore" by switching immediately from her initially selected arm to the alternate arm, returning to the initial arm if that alternative arm pays 0, and "exploiting" the alternative arm (playing it forever) otherwise.[1]

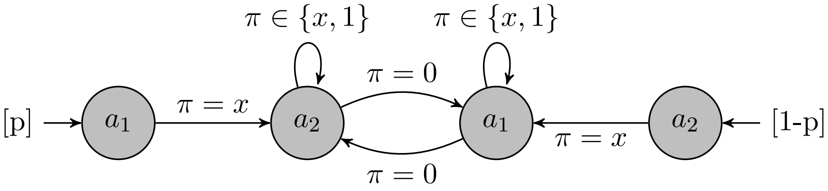


Figure 1: Automaton representation of an optimal 4-state procedure.

[1] This variation on the bandit problem and the automata analysis of it discussed below are adapted from Börgers & Morales (2004).

Figure 1 visualizes the finite automaton representation of one form of the optimal procedure (the optimal decision rule or decision algorithm) for this problem. Each of the circles represents a *state*. Each state (i) specifies an action (choose arm $a_1$ or $a_2$, shown inside each circle) and (ii) gives rise to a set of transitions (shown as arcs emenating from the state) to the next period's state as a function of events (here, the payoff $\pi$ received after each action, shown next to each transition). Finally, free standing arrows pointing at the states at the extreme left and right of the diagram show the two initial states the decision maker (DM) can choose from by pulling one of the two initial arms. $p$ and $[1-p]$ next to these arrows represent the probabilities with which the agent randomly chooses either initial arm. This procedure instructs the DM to randomly choose either arm (given the payoffs, either $a_1$ or $a_2$ is equally good, and this choice determines whether the initial state is to the far left or far right); immediately switch to the other arm upon receiving the initial payoff of $x$ (shown as transitions pointing inward from each of the two initial states); and remain on that arm (the reflexive transition looping above each of the second states) if the payoff there is $x$ or 1, but instead advance to the third state from the initial one if it pays 0. This third state instructs the DM to choose the arm she initially selected, forever.

The core idea of the automata literature is that people dislike implementing *complex* procedures, and that this aversion drives procedural choice. Thus, the literature focuses on what it calls "implementation complexity" (the subjective cost of implementing a mentally laborious procedure) rather than "computational complexity" (the difficulty of determining what procedure is optimal). The literature usually operationalizes implementation complexity by assuming it is generated by the number of *states* in a procedure ("state complexity" or *s-complexity*). The idea is that every state added to a procedure increases the burden of implementing it by requiring the decision maker to pay richer attention to history and to process information in more distinct ways over the course of the problem.

Although the optimal procedure described above is not complex to compute, it *is* complex to implement (requiring, as it does, four states). As Börgers & Morales (2004) show, a decision maker can economize on implementation complexity by substituting to a simpler (lower state) procedure instead. For instance, by making arbitrary decisions (e.g. playing randomly in all histories) the decision maker can remove three states from the rule. By suboptimally choosing not to explore at all or by selectively (and again, suboptimally) randomizing in some histories she can remove two states. Finally, by *not* using a general form of the rule and instead "hard coding" an initial arm choice into the rule itself (i.e. by using a "hardcoded" rule, built on a long-run habit), she can remove one state from her procedure without sacrificing any earnings.[2]

The key idea behind our experimental design is that we can ease or exacerbate the costs associated with implementing state-complex procedures by varying the mental effort subjects must exert to track states. In our baseline State Tracking or "ST" treatment, we require subjects to bear the

[2] Empirically, we can detect hardcoded procedures by having subjects encounter the task multiple times: a subject using a hardcoded rule will always select the same initial arm, each time she faces the task.

full burden of tracking states themselves. Specifically, subjects must keep track of what actions they have taken and what payoffs they have observed in the past and condition their actions on these events. We also remove subtle mnemonic aids in the display that subjects might use to reduce the costs of state tracking: by scrambling the keys used to select actions and the arrangement of information on the screen, we remove external crutches to memory, exposing subjects to the full cost of tracking states. By contrast, in our No State Tracking or "NST" treatment, we artificially *reduce* the costs of tracking states by having the computer track and organize information about both past actions and the findings from past exploration. This treatment variation is designed to target the cost of tracking states: other standard primitives about the problem (payoffs, risks, and, to the degree possible, instructions) is kept identical. Under the hypothesis that procedural choice is governed by the costs of state-complexity, we should observe subjects choosing systematically lower-state procedures in State Tracking than in No State Tracking.

We find striking evidence in favor of this hypothesis. Despite the simplicity of the decision problem, fewer than half of ST subjects behave optimally in the typical bandit task. This suboptimality all but disappears when we remove the costs of tracking states in the NST treatment, with optimization rates rising to as high as 90%. Conducting a detailed analysis of individual decision making, we find that most subjects (about 90%) use consistent and identifiable procedures in both treatments. Most ST subjects use procedures with fewer than four states, with many using hardcoded 3-state procedure or 1-state "non-exploration" procedures. A number of ST subjects simply ignore their payoffs and make arbitrary choices, using a minimally complex 1-state procedure. In NST, by contrast, most subjects use maximally complex 4-state procedures and virtually none use simple 1-state Non-exploration or Non-tracking procedures. Of particular diagnostic importance, we find that even among subjects that *do* optimize, NST and ST behaviors are dramatically different. Subjects in NST only rarely bother to use habitual, hardcoded first arm-choices that economize on states, while almost all optimizing subjects do so in ST. This is particularly strong evidence that subjects deliberately economize on complexity costs, and are capable of doing so in a relatively sophisticated fashion.

Crucially, our results strongly suggest that these patterns are not accidents due to changes in comprehension or mistakes (e.g. slips of the hand), but are rather due to subjects *deliberately* attempting to economize on complexity by formulating and substituting to simpler procedures. For instance, the results do not arise because of confusion: the subset of subjects who score almost perfectly in a comprehension quiz also behave in systematically different fashions in ST and NST. The design also allows us to reject accidental mistakes generated by task difficulty as an explanation of our findings: most simplifying procedures require highly patterned actions that cannot plausibly arise by chance for decision makers attempting to behave optimally. In order to study this further, we ran an additional diagnostic treatment that does not target the cost of states, but does make the problem generically more difficult. In our State Tracking + Distraction (ST+D) treatment, we repeat the ST treatment but require subjects to conduct a simultaneous task, causing decision making to become harder (subjects take twice as long to make decisions in this treatment). We find that, in contrast to NST, this has no systematic effect on the procedures subjects use relative to ST, suggesting that our main results arise from the specific channel (aversion to complexity) hypothesized in the automata literature. Our design also immediately rules out a number of alternative explanations like risk or other uncertainty preferences, because the risk characteristics of the problem is identical in ST and NST.

Finally, in an additional diagnostic treatment called NST-ST, we show that the same patterns occur within-subject. In this treatment we first run subjects through a number of tasks under the NST treatment (suppressing state complexity costs) and then increase complexity costs by running them through a number of tasks under the ST treatment. We find that subjects, on average, respond to this "shock" to state complexity costs by transitioning to simpler (lower state) procedures. Subjects in the NST-ST treatment are five times as likely to make this transition to simpler procedures than subjects in the NST treatment who never experience this cost shock. This is, again, strong evidence that complexity is a major driver of behavior.

It should come as little surprise to economists that people deviate from optimality more often when it is costly to do so. But, importantly, this is not what our experiment was designed to study. Rather, it was designed to evaluate a distinctive set of hypotheses from the automata literature about (i) how these costs arise in the algorithmic structure of procedures and (ii) decision makers' sophistication at articulating and deploying procedures to avoid complexity costs. First, the experiment shows, causally, that people find the central aspect of behavior described in the automata literature – tracking states – costly enough to motivate avoidance of optimal procedures. Second, the experiment shows that subjects are sensitive enough to these costs and sophisticated enough at avoiding them, to reliably construct alternative low-state procedures instead. Finally, and perhaps most importantly, the experiment shows that this type of complexity has large effects on behavior even in an extremely simple dynamic decision making context. The fact that subjects deviate from optimality to avoid state-complexity in what is perhaps the simplest possible infinite horizon setting suggests that this type of complexity may be of first order predictive and explanatory value in a much wider range of contexts.

Our findings provide, perhaps, the most direct empirical evidence to date in favor of the "procedures and complexity" approach to bounded rationality, as formalized in the automata literature. Despite the promise of this approach as a way of integrating bounded rationality into economic theory, it has received surprisingly little empirical attention thus far.[3] In a related paper, Oprea

[3] Although there is relatively little evidence on the effects of algorithmic complexity on procedural choice, there is a literature on other facets of bounded rationality that is connected. For instance, recent experiments have measured the attentional costs of complexity in inference problems (e.g. Caplin, Csaba, Leahy & Nov 2019, Dean & Neligh 2019, Carvalho & Silverman 2019, Dewam & Neligh 2020), the effects of complexity on rational choice and search (e.g. Caplin, Dean & Martin 2011, Gabaix, Laibson, Moloche & Weinberg 2006, Sanjurjo 2017) and the effects of the uncertainty produced by complexity (Enke & Graeber 2020). Alaoui & Penta (2016) provides evidence that thinking costs explain findings of limited depth of reasoning in strategic problems. Halevy & Mayraz (2020) and Nielsen & Rehbeck (2020) provide evidence (in the domain of lottery choice) that people prefer to make decisions using procedures/rules.

(2020) exogenously assigns subjects artificial rules (meaningless algorithms specifying sequences of letters that must be typed in order to earn payoffs) and elicits their distaste for implementing these rules again in the future. That paper finds that subjects are averse to several automaton characteristics (including, notably, states) but does not assess, as we do, the effect of complexity costs on actual *behavior* in a real choice settings. Unlike Oprea (2020), we observe subjects' own endogenous formulation of and choice over procedures and we thus are able to study the degree to which automata models actually organize behavior in a canonical economic environment. Also related is Jones (2014), who compares simple prisoner's dilemmas to multi-state dynamic games; in the latter case the number of automaton states required to cooperate is higher. Jones (2014) finds evidence that subjects cooperate less often in the latter case, providing indirect evidence of aversion to state-complexity. By studying much simpler bandit problems, we are able to directly identify the procedures subjects use. Moroever, we are able to more clearly identify state complexity as the source of behavior by using a more targeted intervention that leaves the nature of the decision problem otherwise unchanged.[4]

A secondary contribution of our paper is to provide a detailed analysis of how people behave in bandit problems and why. Bandit problems are canonical in social and economic life, as they describe the fundamental tradeoffs, arising in a wide range of settings, between exploring (learning about the value of options by experimentation) versus exploiting (taking advantage of what has been learned in the past). Here, too, there is surprisingly little prior work given the importance and canonicity of the problem. Using a range of different bandit problems Banovetz (2020), Banks et al. (1997), Hoelzemann & Klein (2019), Gans et al. (2007), and Meyer & Shi (1995) find evidence of over-switching among many of their subjects, a pattern we also find in many subjects in our data. Anderson (2012), Hudja & Woods (2019) and Banovetz (2020) find evidence of a tendency towards under-exploration just as we do. Finally, like us, Gans et al. (2007) and Meyer & Shi (1995) find evidence that subjects employ suboptimal decision making routines (in their cases, systematic overweighting of recent information and/or myopically discounting the distant future). Unlike this previous literature, we use automata to formally benchmark the complexity of the procedures used by subjects and exogenously vary the cost of this complexity to evaluate its causal (and, as it turns out, fundamental) role in determining behavior.

The remainder of the paper is organized as follows. In Section 2 we describe the bandit problem we implement in the experiment, introduce automata models, and describe and operationalize the set of procedures available to economize on state complexity. In Section 3 we describe the experimental design. In Section 4 we provide our main empirical results and we conclude with a Discussion in Section 5.

[4] Automata have also proved useful in recent experimental work for taxonomizing and estimating strategies in repeated games. This literature uses automata to measure and taxonomize strategy choices in repeated prisoner's dilemmas either by using automata in specifying estimators (e.g. Engle-Warnick, McCausland & Miller 2007, Dal Bó & Fréchette 2011) or by allowing subjects to choose or program automata directly when playing games (Dal Bó & Fréchette 2019, Romero & Rosokha 2018, 2019, Cason & Mui 2019).

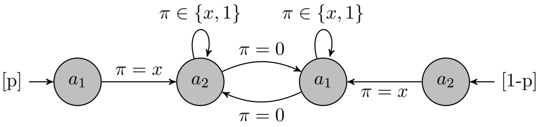
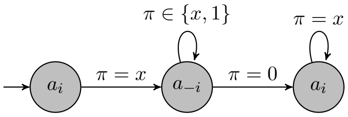
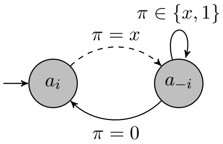
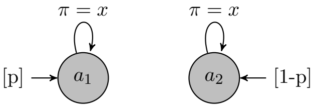
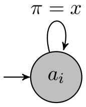
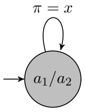
| Rule | Automata | 1A-I | 1A-0 | 2A-1 | 2A-0 | H |
|---|---|---|---|---|---|---|
| **Optimal, Randomized**<br>*4-state* |  $\pi \in \{x,1\}$<br>$\pi \in \{x,1\}$<br>$\pi = 0$<br>[p]<br>$a_1$<br>$\pi = x$<br>$a_2$<br>$a_1$<br>$\pi = x$<br>$a_2$<br>[1-p]<br>$\pi = 0$ | 1 | 0 | 0 | 1 | No |
| **Optimal, Hardcoded**<br>*3-state* |  $\pi \in \{x,1\}$<br>$\pi = x$<br>$a_i$<br>$\pi = x$<br>$a_{-i}$<br>$\pi = 0$<br>$a_i$ | 1 | 0 | 0 | 1 | Yes |
| **Mixed**<br>*2-state* |  $\pi \in \{x,1\}$<br>$\pi = x$<br>$a_i$<br>$a_{-i}$<br>$\pi = 0$ | $y$ | $y$ | 0 | 1 | - |
| **Non-explore, Randomized**<br>*2-state* |  | 0 | - | - | - | No |
| **Non-explore, Hardcoded**<br>*1-state* |  | 0 | - | - | - | Yes |
| **Non-Tracking**<br>*1-state* |  | $\approx 0.5$ | $\approx 0.5$ | $\approx 0.5$ | $\approx 0.5$ | - |

Table 1: Automaton representations of six procedures. *Notes: The right five columns give (i) the signature arm-switching probabilities for each procedure from a set of diagnostic histories and (ii) in the final column whether the procedure is hardcoded.*

## 2 Conceptual Background

In this section, we introduce the bandit problem we implement in our experiment (Section 2.1) and provide an overview of the use of finite automata to model decision procedures (Section 2.2). In Section 2.3 we describe the set of procedures/automata available to decision makers in our bandit problem and in Section 2.4 we show how to identify these procedures in observed behavior. Finally in Section 2.5 we discuss the interpretation of states and state complexity.[5]

[5] Much of the material in this Section is inspired by Börgers & Morales (2004) and we refer the reader there for more details.

### 2.1 A Bandit Problem

Consider the following simple bandit problem analyzed by Börgers & Morales (2004) and adapted for use in our experiment. In each period $t$, a decision maker must choose ("pull") one of two actions ("arms") $a_i \in \{a_1, a_2\}$. The first arm the DM chooses (whichever arm it is, hereafter called the "first arm") is guaranteed to pay a fixed amount $x$ each time (in each period) it is selected. By contrast there is uncertainty about the (also fixed) payoff of the alternative arm (hereafter called the "second arm"): ex ante, the second arm will pay out 0, $x$ or 1, each with equal probability. Each period there is a probability $(1 - \delta)$ that the game ends.[6]

It is optimal to *explore* in this problem by immediately switching away (in the second period) from the first arm, as long as $x$ is smaller than a critical value, equal to the stream of expected payoffs from exploring in an optimal way. If the DM "explores" by switching to the second arm, she should optimally (i) *exploit* the second arm by playing it forever if its payoff is 1, but should instead (ii) return to the initial arm and play it forever if the second arm's payoff is 0 (if the second arm's payoff is $x$ then it does not matter what arm the DM chooses afterwards). Thus it is optimal to *explore* by immediately switching away from the initially selected arm if:

$$\left(\frac{x}{1-\delta}\right) \leq \frac{1}{3}\left(\frac{\delta x}{1-\delta} + \frac{x}{1-\delta} + \frac{1}{1-\delta}\right)$$

or

$$x \leq \frac{1}{2-\delta}.$$

We will assume this condition holds in what follows (and parameterize the experiment so that it holds in our experimental design).

### 2.2 Automata and Procedures

Suppose a decision maker (DM) is guided by a procedure (a rule) that specifies (i) the actions $A = \{a_1, a_2\}$ the DM will take (ii) in response to payoff *events*, $E = \{0, x, 1\}$ (payoffs received from the action most recently taken), and, crucially, (iii) how the mapping between payoffs and actions is conditioned on history.

Following the automata literature, we describe such decision rules formally using *finite automata*, simple models of algorithms imported from computer science (e.g. Hopcroft & Ullman (1979)). An automaton (a *finite state machine*) is a four-tuple $(S, s^0, f, \tau)$ where $S$ is a set of *states*, $s^0$ is the set of initial states, $f : S \to R$ is an output function, designating an action in each state, and $\tau : S \times E \to S$ is a *transition* function that selects a next state as a function of the current state

[6] Börgers & Morales (2004) study a version of the problem in which there is ex ante uncertainty about both arms.

and the current event. Each row of Table 1 represents an automaton visually in the standard way, as a directed graphs with (i) circles representing states, (ii) arcs representing transitions, (iii) symbols inside of the circles representing actions from a set $A = \{a_1, a_2\}$ and (iv) expressions next to transitions representing events from a set $E = \{0, x, 1\}$. Free standing arrows pointing at states designate the possible initial states for the rule, determined by the DM's initial action. When multiple initial states are possible in a rule, the agent randomizes her initial actions and probabilities (e.g. $[p]$, $[1-p]$) are listed next to initial arrows to parameterize these probabilities.

The top row of Table 1 shows an automaton for a randomized version of the *optimal rule*. The DM first randomly chooses an initial arm, thereby entering either the state to the far left (if she initially choose $a_1$) or right (if she initially choose $a_2$) of the graph. The arrows pointing inward from each of the initial states each instruct the DM to immediately move to a new state, choosing the second (non-initial) arm if she receives a payoff of $\pi = x$ on the first arm (which, recall, in this version of the problem, she is guaranteed to receive on her initial arm, regardless of which arm this is). That is, the rule instructs the DM to immediately *explore* after her first choice. If the DM receives a payoff of $x$ or 1 on the second arm, the rule instructs her to remain in this state (this is represented by the reflexive, looping transitions above the each of the two inner states). If, however, she receives a low payoff of 0 on the second arm, she transitions one more state away from her initial state and, given the structure of payoffs, is instructed to remain at that state in every future period.

This rule requires four states to accomplish two key requirements of the optimal procedure:

1. **Track past actions**: The rule requires one state for each of the DM's two possible initial actions (the far left state corresponding to $a_1$ and the far right one corresponding to $a_2$), allowing the DM to track which arm she chose initially.

2. **Track past events**: The rule requires a second state for each of the two possible initial actions to distinguish between histories in which she has not yet experimented and histories in which she has experimented and observed a low payoff on the second arm.

Thus, the states in the randomized version of the optimal rule allow the decision maker to track and condition further behavior on what she has done and what she has learned in the past.

## 2.3 Complexity and Simple Procedures

The automata literature studies how procedures vary in their *complexity* and it generally defines the complexity of a procedure as the number of *states* it contains (sometimes called state complexity or "s-complexity"). A DM that is averse to complexity will *ceteris paribus* prefer rules with fewer states, and may even be willing to sacrifice earnings to use rules that economize on states if she is sufficiently averse to complexity.

A costless way of reducing complexity in our bandit problem is to avoid randomization by "hard coding" an initial action into the logic of the rule: each time the DM faces the decision problem, she uses a rule that always begins with the same initial arm (which can be either $a_1$ or $a_2$, arbitrarily, given our payoffs). By doing this, the DM can simplify the procedure she deploys in the problem: intuitively, the DM no longer has to track her initial choice, which removes a state from the rule. As the second row of Table 1 shows, this hardcoded version of the optimal rule allows the DM to reduce the rule's state-count at no cost simply by establishing a long-run habit of using a consistent initial action. The DM now always plays $a_i$ initially (with $i$ fixed across every deployment of the rule), immediately explores by playing $a_{-i}$, and transitions to a terminal state specifying $a_i$ again if $a_{-i}$ turns out to pay zero. This rule is simpler than the randomized version of the optimal procedure, but is nonetheless optimal: because $a_1$ and $a_2$ are symmetric ex ante, there is no cost to habitually choosing one of the arms initially and building a simplified rule around this habit.[7]

A costlier but more effective "tool" for avoiding complexity is to ignore past information, by processing information (payoffs) at each arm identically regardless of what has occurred in the past. If the DM is willing to forget whether she has observed a 0 payoff in the past and randomize her arm-switching behavior every time she observes a payoff of $x$, she can implement the *2-state* **mixing** procedure shown in the third row of Table 1. This rule specifies that the DM start with some action, $a_i$ and transition from that initial action to $a_{-i}$ with some probability $q$ (the dotted transition line from the first state specifies a *stochastic* transition, made with some iid probability). At the second arm, if she receives 0, she returns to the initial action $a_i$ immediately, but does not keep track of this history (note that if she earns 1 from the second arm, she will remain on that arm forever). Instead, she transitions once again from the first arm each period with probability $q$. Randomization allows the DM to reduce states, and if $q$ is well-calibrated, she can do this in a minimally costly way (in our parametrization in which $x = 0.65$ the optimal choice of $q$ in a 2-state rule is 0.19).[8]

More radically still, the DM can simply avoid exploration altogether, employing a **non- exploration** procedure. In our setting, this guarantees the DM the known intermediate payoff of $x$, and reduces the procedure to a 2- or 1-state rule. The "Non-explore, Randomized" row in Table 1 shows the 2-state non-exploration automaton which instructs the DM to randomly choose an initial arm ($a_1$ or $a_2$), determining her initial state (left or right), and then to continue choosing that arm throughout the task. This procedure contains two states because it requires the decision maker to remember which action she initially selected. The decision maker can go further by implementing a

[7] Although hardcoding doesn't impact payoffs in our task, it nonetheless serves as a very valuable (and very empirically distinct) barometer of subjects' sensitivity to state-complexity and sophistication at avoiding it when constructing and choosing between procedures.

[8] In our bandit problem (in contrast to the one studied in Börgers & Morales (2004)), this 2-state randomization procedure can be implemented with or without hardcoding of the initial action. The automaton pictured in Table 1 is a version with initial bias, but a different 2-state randomization procedure can be constructed with no hardcoding but with several additional transitions between states. In 1 we list the 'H' prediction as '-' to acknowledge that either characteristic is consistent with this type of 2-state procedure.

hardcoded version of non-exploration, forming a cross-game habit of always selecting (and forever after always playing) the same initial arm $i$. Doing this reduces non-exploration to a 1-state rule (pictured in the "Non-exploration, Hardcoded" row), which specifies choosing the same arm within and across instances of the problem, again guaranteeing a payment of $x$.

Finally, a decision maker can avoid state complexity by randomizing which arm she selects in each period with equal probability. This **non-tracking** rule, pictured in the final row of Table 1, requires only one state (the action in that state is listed as "$a_1/a_2$," indicating that the DM randomizes between the two actions). This rule is simple, but is the rule that yields the lowest payoff.

Removing states from procedures – moving between 4-state, 3-state, 2-state and 1-state – progressively reduces state-complexity, but also weakly reduces expected payoffs: 4-state and 3-state are payoff equivalent, with payoffs decreasing as states are shed and minimizing with the non-tracking 1-state rule. Thus, subjects face a trade-off between complexity costs and monetary earnings in choosing between procedures.

## 2.4 Behavioral Fingerprints of Procedures

Each of the procedures discussed in the previous section produces a distinctive set of behaviors that we can observe in decision making. To identify procedures, we classify a decision maker's history at the moment of choice according to two criteria:

- **Preceding Action**: The DM has either just selected the first arm (the arm she selected in period 1, which we will code as "1A") or the second arm (the arm she did not initially select, coded "2A").
- **Past Events**: The subject either (i) has not yet explored (coded "I" for "initial") or (ii) *has* explored (has selected the second arm at some previous point in the task) and learned that it pays 0 (coded "0"), $x$ (coded "x") or 1 (coded "1").

The six procedures in Table 1 differ in how frequently (with what per-period probability) they instruct the DM to "switch" arms (abandon their current, e.g. most recently selected, arm and play the other arm instead) in each of four key diagnostic histories. We label these (combining codings from the criteria listed above):

- **1A-I**: the probability with which the DM "explores" by switching away from the first arm when she has not yet tried the second arm
- **1A-0**: the probability with which the DM switches away from the first arm, after having previously learned that the second arm pays 0

- **2A-0**: the probability with which the DM switches away from the second arm after it pays 0
- **2A-1**: the probability with which the DM switches away from the second arm after it pays 1.[9]

Procedures are also differentiated by whether they require the agent to make use of “hard coding,” by consistently choosing the same first arm over repetitions of the problem.

In the five right hand columns of Table 1 we list the per-period probability with which the DM is predicted to switch away from her most recently selected arm in each of these histories, and whether the procedure requires the DM to employ hardcoding (‘H’). We highlight several key points.

First, all procedures (other than the final “non-tracking” rule) include a basic rationality requirement: that the DM immediately switch away from the second arm if it pays 0 (2A-0= 1), and never switch away from the second arm if it pays 1 (2A-1= 0). Following these requirements only requires the DM to track the action she took in the previous period (or, in the case of the hardcoded non-exploration procedure, to remember the unique action hardcoded into the rule) and to condition her next action on that action’s payoff. We will call failures to follow this minimal pair of requirements taking “second arm dominated” actions. The only procedure that specifies the DM should take such dominated actions is the 1-state “non-tracking” procedure. Indeed, since this procedure requires the DM to pay no attention to prior choices or payoffs, following it should produce similar, strictly interior switching rates in all four histories (1A-I≈IA-0≈2A-1≈2A-0≈ 0.5).

Second, at the opposite extreme, decision makers using 3- or 4-state Optimal procedures will (i) explore at a high rate (1A-I= 1), (ii) avoid taking second arm dominated actions (2A-0= 1 and 2A-1= 0), and (iii) avoid switching back to the second arm if it has paid 0 in the past (1A-0= 0). Decision makers using the 4-state randomized Optimal procedure will choose initial actions randomly (hardcoding, H = ‘No’), while those using 3-state hardcoded Optimal procedures will consistently choose the same initial arm (either $a_1$ or $a_2$) each time she encounters the task (H = ‘Yes’).

Third, decision makers using 1- or 2-state Non-Exploration procedures will rarely explore (1A-I= 0). These procedures do not put constraints on off-path play in histories 1A-0, 2A-0 or 2A-1, though subjects using Non-Exploration will produce little data from these histories in any case. Decision makers using 2-state randomized Non-Exploration procedures will choose initial actions arbitrarily (H = ‘No’), while those using 1-state hardcoded Non-Exploration procedures will consistently choose the same initial arm (either $a_1$ or $a_2$) each time she encounters the task (H = ‘Yes’).

[9] Several other histories, of course, can be constucted but do not produce distinctive predictions across procedures. In 1A-x and 2A-x, decision makers should be indifferent across arms (because these histories can only occur when the two arms pay the same amount). 1A-1 should not occur for most of our procedures, and so we can’t use it to “fingerprint” procedures.

Finally the 2-state Mixed procedure requires the DM to (i) avoid taking second arm dominated actions (2A-0= 1 and 2A-1= 0) but to "forget" what she has learned in the past about the second arm (or even whether she has played the second arm). That is, whenever playing the first arm, a DM using this procedure should switch to the second arm with an equal probability $q$ *regardless of history* (1A-I=IA-0= $q$). (Given our parameters a constrained optimally calibrated Mixed procedure will set $q = 0.19$.)

## 2.5 What is State Complexity?

We will follow the automata literature, by considering the hypothesis that the number of states in a procedure is an important measure of the complexity of that procedure. Oprea (2020) measures the amount subjects are willing to pay to avoid implementing procedures describable as automata and shows that, indeed, adding states to a procedure generates direct cognitive costs for subjects. Our experiment, described below, will test the hypothesis that this type of complexity meaningfully influences procedural choice. Before describing and motivating the experiment, it is useful to reflect on what a "state" in an automaton is, psychologically. What is it exactly that a decision maker economizes on when she chooses to use a lower-state procedure?

In a finite automaton, a state combines two main computational demands into one object. First, a state summarizes the information processing required by the procedure. That is, it describes the number of distinct ways the decision maker must interpret and respond to new events over the course of the task. When a procedure includes more states, the decision maker must manage more contingencies in order to comply with that procedure and there are therefore more distinct types of mistakes she must exert effort to avoid.[10] Avoiding these mistakes in attempting to implement a multi-state procedure requires cognitive control, the key hypothesized driver of cognitive costs in recent work in cognitive science ((e.g. Kool & Botvinick 2018, Shenhav, Musslick, Lieder, Kool, Griffiths, Cohen & Botvinick 2017)). Second, a state is the way an automaton encodes the *episodic memory* required of the rule – the events from the past that must be recalled to comply with the procedure's instructions. This demand on memory may bring distinct costs of its own.[11]

Which of these aspects of a state generate complexity costs for humans? Oprea (2020) provides evidence that it is largely the information processing aspect that subjects are willing to pay to

[10] See Proto et al. (2022) for an analysis that emphasizes that mistakes in implementing procedures arise due to failures to properly transition between states. As states are added to a rule, the number of such errors that must be avoided rises.

[11] Finite automaton are relatively "coarse" descriptions of algorithms – they lie near the top of the "Chomsky hierarchy" that linguists and computer scientists use to classify the descriptive sophistication of algorithms. Descending the hierarchy, there are richer classes that separate out the "information processing" and "memory" components of a rule; in finite automaton the two are combined. One step down in the hierarchy are "pushdown automata" which work like finite state machines but can additionally respond to events by writing, reading and deleting symbols to a string held in working memory. In some types of automata this separates the episodic memory requirements of a procedure from the information processing requirements (see Oprea (2020) for details).

avoid. He shows this in two ways. First, in one of his treatments subjects are allowed to see the full history of past events while in the other subjects only see the most recent events. There is no difference in elicited costs between these treatments, suggesting that episodic memory has little to do with these costs. Second, he compares subjects' willingness to pay to avoid implementing two structurally identical automata. One, an automaton called "4S-4T", uses states mostly for information processing: each state contains unique instructions on how to respond to events. Another, an automaton called "countable," by contrast, uses three of its states merely to track how many times a prior event has occured, using states largely for memory. Subjects are willing to pay substantially less (the equivalent of two states less) to avoid being assigned "countable" than "4S-4T." This, too, strongly suggests that information processing is a dominant driver of state complexity costs.

# 3 Experimental Design

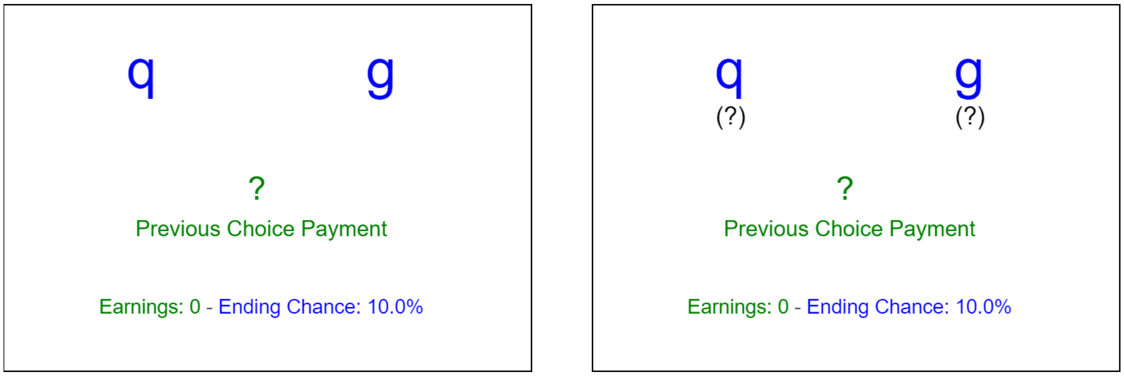
The experiment consists of 20 *tasks*, each of which is a full, multi-period implementation of the bandit problem described in Section 2.1. The subject chooses between pulling two arms, 1 and 2. To make the problem easier to understand, we scale payoffs by a factor of 100. The first arm pulled (note that this could be arm $a_1$ or $a_2$) always pays $x = 65$ points. The second pays 0, 65 or 100 points, each with equal probability. Each period there is a $(1 - \delta) = 0.1$ chance that the period is the last.[12] As we show in Online Appendix A, this choice of $(x, \delta)$ guarantees that the optimal procedure is the one described in Section 2 (i.e, a payoff-maximizing subject should always explore).

Figure 2 shows a pair of screenshots from the experimental software that illustrate how we implemented the task. In each period, subjects selected an arm of the bandit by typing a letter linked to that arm. The letters on the screen changed randomly each period, but subjects knew that typing the alphabetically earlier letter (of the two on the screen) always selected action $a_1$ and typing the alphabetically later letter selected $a_2$ The two letters required for selection were also shown in a random left-right orientation on the screen each period. We implemented these two features (random letters to select arms and random screen orientation) to reduce the ability of subjects to rely on artificial mnemonic devices otherwise available in any computerized implementation.[13] This gives us greater control over the costs of complexity by forcing subjects in our main treatment to bear the full burden of tracking states.[14] After pulling an arm by typing a letter, subjects were

[12] In practice we randomly pre-drew the number of periods for each task ahead of time. The mean task lasted 11.4 periods and the longest lasted 29 periods.

[13] For instance, if the left-right orientation were held constant across periods, the subject could focus her eyes (or rest her finger) on a side of the screen to track actions that have been taken in the past. To remove this prop we randomized which option appeared on the screen. Another mnemonic is to rest one's finger on a key representing the action the subject took in a previous period, aiding memory – changing the letter required for each action in each period prevented this. Together, these force the subjects to track which arms they have selected in the past and what their payoffs were in each case.

[14] Making use of mnemonics like placing a finger on the screen or resting it on the keyboard can effectively remove


q g

?

Previous Choice Payment

Earnings: 0 - Ending Chance: 10.0%

q g

(?) (?)

?

Previous Choice Payment

Earnings: 0 - Ending Chance: 10.0%

(a) ST

(b) NST

Figure 2: Screenshots for the two main treatments.

shown their latest payoff, their cumulative earnings and were given a new pair of letters to input their next choice.

We ran the experiment using the *block random termination* (BRT) protocol introduced by Frechette & Yuksel (2017). Subjects made their choices in blocks of five periods and were only told if and when the final "real" (paying) period of the task had occurred once a block was over. This protocol retained the intertemporal discounting incentives of the model but allowed us to gather more periods of incentivized data per task. Once a task was completed, a subject began a new task until the subject had played 20 total tasks.

### 3.1 Treatments: Varying the Costliness of Complexity

The heart of the experimental design is a pair of treatments, applied between-subjects, that vary the complexity costs of tracking states without changing the formal properties of the decision problem. In our **State Tracking** treatment (ST), visualized in Figure 2a, we implemented the experiment in the natural way: subjects were required to track both their history of actions and their prior payoffs themselves, thus exposing them to the two key sources of state-complexity in the problem. In the **No State Tracking** (NST) treatment we removed the need for subjects to track states themselves, as Figure 2b shows. The display is identical to the one from the ST treatment except for one change: subjects are shown question marks ('?') below each of the action-letters, which are replaced with the payoff of each arm once the subject has sampled that arm. Subjects in NST therefore did not have to track past events and choices to implement any of the rules in Table 1. This removed the cognitive burdens summarized by state-complexity, while keeping all other features of the problem constant.

We also included two diagnostic treatments in the design, motivated and discussed in Sections

the cost of tracking one state in several of our procedures. By removing subjects' ability to easily do this, we therefore exert stronger experimental control on the cognitive costs associated with implementing procedures.

4.4 and 4.5. In treatment State Tracking + Distraction (ST+D) we add a distraction task to the ST treatment in order to increase task difficulty without altering the costliness of tracking states. In treatment NST-ST, we assign subjects ten tasks under the NST treatment followed by ten tasks under the ST treatment in order to study how subjects respond to a state-tracking cost "shock."

### 3.2 Implementation

We ran the experiment using custom Javascript software, deployed via Qualtrics. Participants were recruited using the Prolific platform (prolific.co) in April and May, 2020. We recruited 180 subjects from the United States (60 in each treatment) and subjects were allowed to participate in no more than one session/treatment.[15] After reading instructions (reproduced in Online Appendix B), subjects were required to take a comprehension quiz consisting of 7-8 questions that we will use to understand the role of confusion in subjects' decision-making. Most sessions lasted in total between 20 and 30 minutes. Subjects earned a $2.50 show-up fee for completing the study and a bonus based on their *average* earnings across all 20 games. Subjects in ST earned an average of $5.05 ($12.13/hour); subjects in NST earned an average of $7.14 ($17.13/hour).[16,17]

### 3.3 Hypotheses

Our first hypothesis is that removing state complexity costs in the NST treatment should lead to higher adoption of more complex (3- and 4-state) optimal procedures than in the ST treatment. This is because ST subjects (unlike NST subjects) can economize on complexity costs by choosing instead simpler 1- or 2-state Mixing, Non-Exploration or Non-Tracking procedures.

**Hypothesis 1** *Subjects are more likely to use optimal procedures in the NST treatment than in the ST treatment.*

A subtler hypothesis, driven by the same complexity-economizing mechanism, is that subjects will also tend to optimize differently in ST than in NST. ST subjects, motivated to economize on states, will show a systematic preference for 3-state over 4-state optimal procedures. NST subjects,

[15] One potential concern about running this experiment online is that subjects may be able to write down what has happened in the past, relieving themselves of the episodic memory demands of the procedure. However, as we discuss in Section 2.5, state complexity describes more than the episodic memory burdens of a procedure. It also describes the information processing and therefore the cognitive control required to implement a procedure. As a result, we should expect state complexity to generate costs even in an online implementation in which subjects can write down past events. Indeed, Oprea (2020) shows that recording and showing the entire history of a task to subjects during the task has no effect on elicited state complexity costs.

[16] Prolific requires researchers to pay subjects at least $6.50/hour.

[17] We paid subjects according to a threshold rule to intensify incentives: if subjects earned fewer than 700 points in the average task, they earned zero and they earned 3 cents for every point greater than 700. This threshold was very easy to meet and in practice 95% of subjects exceeded it.

facing no state-tracking costs, by contrast should be approximately indifferent between the two. For this reason, we expect to see more systematic use of 3-state procedures in ST than in NST.

A subtle feature of our design may intensify this comparative static by breaking indifference, causing NST subjects to *systematically* default to the 4-state Optimal procedure when relieved of state-tracking costs. In particular, our design makes it slightly *easier* to randomly choose an initial arm than to deterministically choose one. In order to deterministically choose an arm in our design, a subject has to exert a very small amount of mental effort to identify which letter on their screen is the lower letter in the alphabet. By contrast, choosing a letter arbitrarily (with no effort at all) *automatically randomizes choice*, because the position of each arm is randomly arrrayed on the screen each period.[18] Thus there are reasons to expect subjects to default to the 4-state procedure when relieved of state complexity costs, giving us an additional opportunity to test for the effects of state complexity on procedural choice.[19] (Note, this same chain of logic applies also to Non-exploration procedures: conditional on using a Non-exploration procedure, NST subjects should default to the 2-state randomized version while ST should be drawn to a 1-state hardcoded version to avoid state complexity costs.[20])

**Hypothesis 2** *Subjects are more likely to use lower-state versions of Optimal and Non-exploration procedures in the ST treatment than in the NST treatment.*

# 4 Results

In this section we report the results from the experiment, focusing on our main ST and NST treatments. In Section 4.1 we give an overview of behavior at the aggregate level. In Section 4.2 we report individual level data and provide evidence on the types of procedures subjects use. In Section 4.3 we formally classify subjects according to the procedures they employ and describe how procedural choice changes over treatments. In Section 4.4 we evaluate the role of task difficulty

[18] For instance, the subject can simply type whichever letter appears on the left side of the screen (or right), or choose whichever letter catches their eye first and this will automatically randomize arm-selection.

[19] To be more precise suppose, due to the ease of randomization, there is a slight excess cost $\varepsilon$ of implementing hardcoded relative to randomized procedures (notice, equally present in NST and ST), in addition to a cost $c$ attached tracking each additional state when such costs are not removed from the design (i.e. in the ST treatment). As long as $\varepsilon < c$, we should expect subjects in ST to prefer 3-state to 4-state Optimal procedures. However, for NST subjects for whom $c$ is removed but $\varepsilon$ remains, we expect to see the reverse in each case.

[20] A useful test of the auxiliary hypothesis that randomized procedures are easier than hardcoded procedures is to look for evidence that subjects in the ST treatment use the 1-state Non-tracking procedure, which allows subjects to randomize while using a minimal-state procedure. While using the 1-state Non-exploration procedure economizes only on state-complexity, the 1-state Non-tracking procedure economizes on *both* types of cost. Evidence of significant use of Non-tracking relative to Non-exploration procedures would not directly speak to our central question (the effect of state complexity on procedural choice), but it would serve as a verification of the premises of Hypothesis 2. From a broader perspective, this additional mental effort cost (though not directly tied to states) is a type of “complexity” and evidence that it has meaningful effects on behavior underscores our general point: that the subjective costs attached to implementing procedures is a significant driver of behavior.

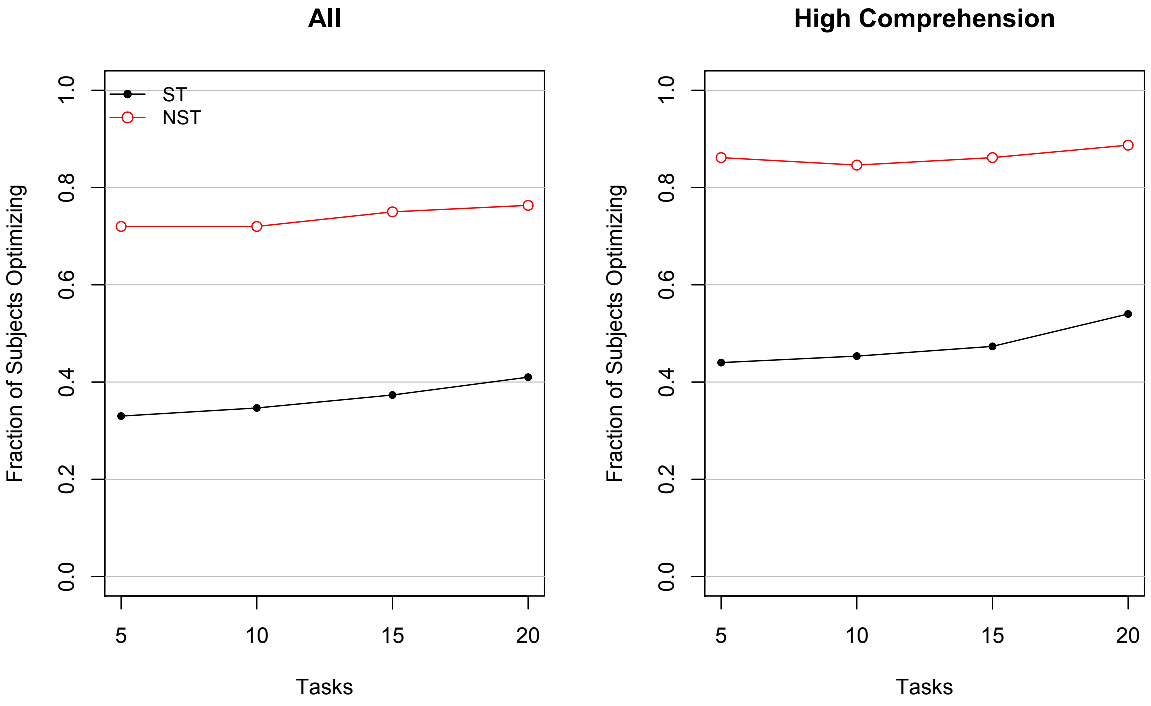


Figure 3: Time series of optimization rates by treatment. *Notes: The plot divides data into four, five-task bins. Each data point is the rate at which subject/task combinations produced an optimal sequence of actions. The right hand panel shows only subjects who made no more than one mistake on their post-instructions comprehension quiz.*

and mistakes in driving our findings and discuss our ST+D treatment. Finally, in Section 4.5 we consider the possibility of differential learning across treatments and discuss the NST-ST treatment.

## 4.1 Suboptimality

We begin by reporting aggregate performance in the bandit task over time and across treatments. Figure 3 divides the 20 tasks into 5-task bins and plots the fraction of tasks in which subjects took an optimal sequence of actions in each bin. The left hand plot shows data for all subjects, and the right hand panel shows subjects that made no more than one mistake in the comprehension quiz at the end of the instructions (we call these “High Comprehension” subjects).

In the baseline ST treatment, a substantial fraction of subjects *fail* to optimize, with fewer than half of subjects playing optimally at almost all points in the session. This failure is costly: subjects sacrifice, on average, about 25% of their optimal earnings by failing to behave optimally. Perhaps more tellingly, ST subjects do a poor job of covering the distance between (i) the earnings of a random decision maker and (ii) those of an optimal player: on average ST subjects only extract 40% of the earnings improvement over random play possible in this problem.

**Result 1** *Less than half of subjects in ST make optimal decisions, leading to substantial shortfalls* *in earnings.*

The data plotted in Figure 3 also tell us much about the mechanism for this failure, casting immediate doubt on several of the most popular explanations for deviations from optimality in the literature. For instance, the results do not seem to be driven by confusion. Subjects (focusing again on ST) show little evidence of learning over the course of the experiment and when we look only at the subjects that show high comprehension in the post-instructions quiz (the right hand panel), we see only a minor improvement in optimization rates. Even after 20 tasks of experience, our least confused subjects still only optimize half of the time.

Because our task is extremely simple, it is also tempting to attribute deviations from optimality to preferences such as risk aversion. However, the data is also inconsistent with this sort of explanation. The NST treatment is identical to the ST treatment in terms of payoffs, randomization and timing.[21] Yet, as Figure 3 shows, almost all subjects behave optimally in NST – among high comprehension subjects, the optimization rate rises to nearly 90% by the end of the session. Moreover, not only is NST identical to ST in terms of mechanics, it is also (with only minor differences) identical in instructions to subjects. The disappearance of most of the deviations from optimality in this treatment therefore also reinforces our rejection of confusion as a primary explanation.

**Result 2** *In contrast to ST, the vast majority of subjects behave optimally in NST.*

By design, the primary thing that differs between these two treatments is the degree to which subjects must exert cognitive effort to track "states" (e.g. past actions and payoffs) themselves when implementing optimizing procedures. Our motivating hypothesis is that subjects are averse to this "state-complexity" and, except when the cognitive burdens of this complexity are removed (in NST), choose to implement systematically simpler procedures instead. In order to evaluate this hypothesis directly, we must examine and compare the procedures individual subjects implement in our bandit problems and across treatments.

## 4.2 Individual Subject Behavior

To identify individual subjects' procedures, we estimate the per-period probability each subject switched arms, $p_h$, in each of the four diagnostic histories, $h \in \{1A\text{-}I, 1A\text{-}0, 2A\text{-}1, 2A\text{-}0\}$, discussed in Section 2.4. For each of the four histories, we calculate $p_h = \frac{1}{\overline{N_h}}$, where $\overline{N_h}$ is the mean number of consecutive periods the subject spent in $h$, over all instances in which $h$ occurred in the dataset. Because tasks end with random probability, it is important that we are careful to minimize bias due to truncation. For our estimates, we therefore include only cases in the sample in which subjects entered the history (e.g. 1A-I or 2A-0) with at least 10 periods remaining in the task, guaranteeing

[21] Of course, it certainly possible that complexity creates a kind of "cognitive uncertainty" (Enke & Graeber (2020)) that generates risk. However, the objective payoffs themselves do not produce differential risk across treatments.

that we can estimate switching probabilities as low as 0.1 (and sometimes lower).[22] We focus our analysis on data from the last half of the session (after period 10), giving subjects a chance to formulate and settle on a procedure before entering the dataset (results are similar if we use the whole dataset instead).

Figure 4 plots estimates of $p_h$ for each history $h$ and each subject in the dataset (each horizontal position is a separate subject). We include separate panels for ST (top panel) and NST (bottom panel). The top plot in each panel shows first-arm switch rates, including for 1A-I (black dot) and 1A-0 (hollow dots); the bottom plot shows second-arm switch rates, including for 2A-1 (black dots) and 2A-0 (hollow dots) for the same subjects. An "H" between plots (on the line labeled 'Hardcoded') signifies a subject who hardcodes her procedure by consistently choosing the same initial arm.[23] Just beneath this (the line labeled 'Quiz') we include the number of mistakes the subject made in the comprehension quiz (e.g., 0 means the subject never made a mistake, 7 means the subject submitted 7 mistaken answers).[24] We order subjects horizontally in Figure 4 first by the median (across tasks) rate at which the subject took second-arm dominated actions, then by the overall per-period rate at which they explored (i.e. by their 1A-I value). Subjects to the right of the first vertical line are subjects who took no second-arm dominated actions in the median period, satisfying our basic test of rationality/attention.

We make several observations.

First, focusing on subjects grouped on the left side of Figure 4, most subjects that make *any* second-arm dominated actions make them *frequently* and under *both* relevant histories (2A-0 and 2A-1). In fact, most of these subjects switch arms with random-looking interior likelihood in *all* four histories (1A-I, 1A-0, 2A-0, 2A-1). This strongly suggests that most of these subjects are not conditioning their actions on their prior choices or payoffs, but are instead choosing arbitrarily, consistent with use of a Non-tracking procedure. We label these subjects as "Non-Tracking" in Figure 4.

In the ST treatment, 42% of subjects in Figure 4 are labeled Non-Tracking, though this proportion shrinks to 27% among high comprehension subjects (subjects who made no more than 1 mistake on their comprehension quiz). Most importantly, when we remove the costs of tracking states in the NST treatment, this behavior virtually disappears, falling to 3%.

**Result 3** *27% of high comprehension subjects in the ST treatment systematically take dominated decisions on the second arm. Most of these subjects switch from all arms with strictly interior*

[22] Recall that the subject does not know how many periods remain because of the stochasticity of the ending rule, meaning this decision introduces no bias due, e.g., to self-selection.

[23] We say a subject displays hardcoding if we can reject the hypothesis at the five percent level that she chooses each arm with equal frequency using a binomial test. In our data this requires the subject to choose the same action in at least 9 out of the 10 tasks in the latter half of the dataset.

[24] Note that subjects could *repeatedly* submit incorrect answers, as the software required a correct answer before subjects could move to the next question. Also note that upon answering incorrectly, the software provided subjects with an explanation that pointed to the correct answer; even so, several subjects produced more than 7 mistakes.

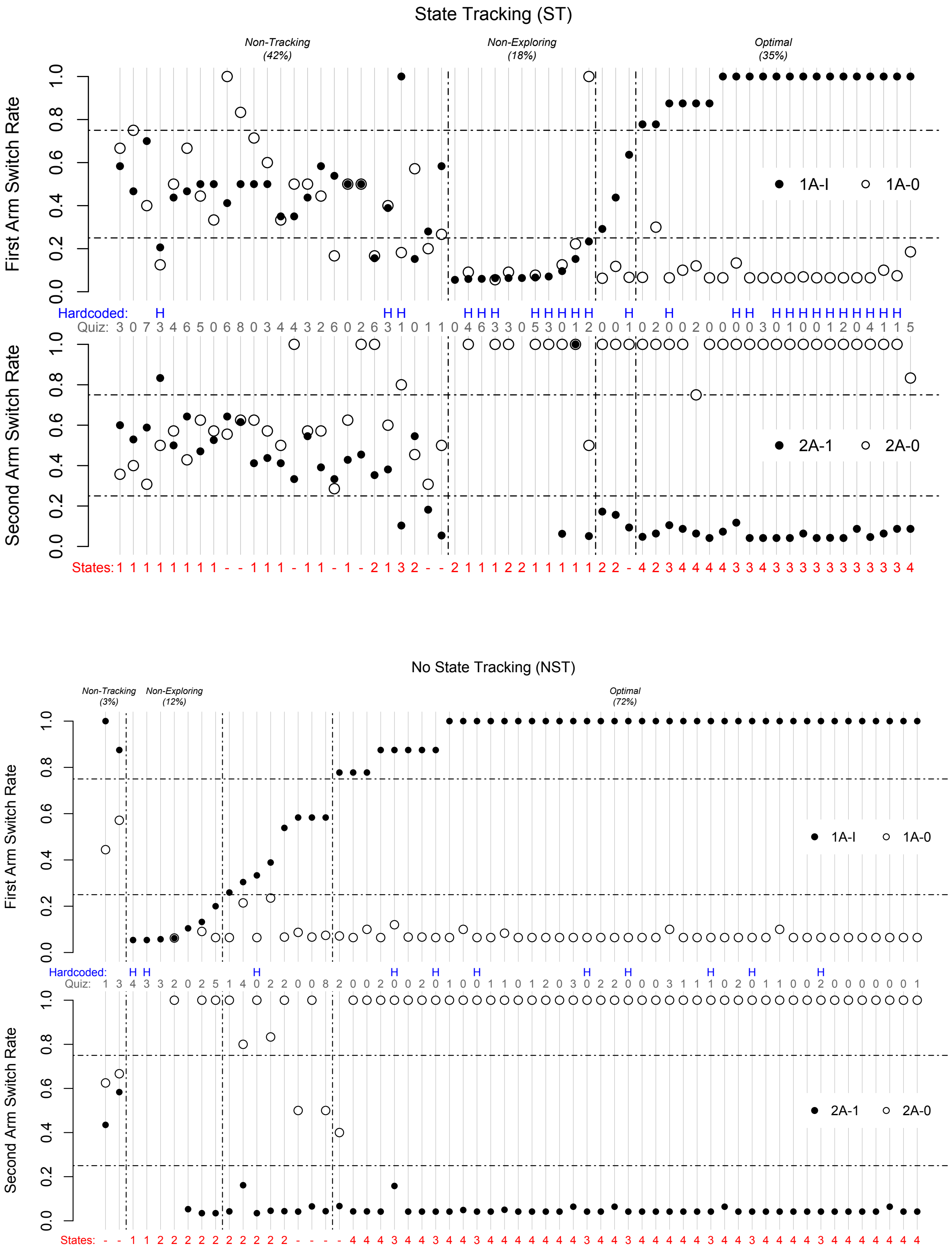


Figure 4: Switching probabilities for each subject in ST (upper panel) and NST (lower panel). *Notes: The upper plot of each panel shows switching behavior after play of the initial arm, the lower plot after play of the second arm. Numbers at the bottom of each panel shows the estimated number of states in the procedure the subject used. Numbers between plots in each panel show the number of of errors the subject made in a comprehension quiz prior to the experiment, and the letter 'H' between plots in each panel designates a subject who made consistent ("hardcoded") initial choices.*

*likelihood, suggesting they are not tracking history at all. This behavior virtually disappears in the NST treatment.*

Second, most of the remaining subjects (i.e. those that do not tend to take second arm-dominated actions in the median treatment, grouped to the right of the first vertical line in Figure 4) either explore with very high probability (e.g. with $p_{1A-I} > 0.75$) or with very low probability (e.g. with $p_{1A-I} < 0.25$). These subjects therefore tend to look much like users of either 1- or 2-state Non-Exploration procedures or of 3- or 4- state Optimal procedures. In Figure 4 we classify these non-dominated subjects as Non-Exploring if $p_{1A-I} \leq 0.25$ or as Optimal if $p_{1A-I} \geq 0.75$ and $p_{1A-0} \leq 0.25$ (though as is clear from the Figure altering these classification boundaries by $+/-0.1$ makes little difference in this classification).

As with Non-Tracking, Non-Exploring is strongly influenced by the cost of tracking states. About 13% of high comprehension subjects (18% overall) are grouped as using Non-Exploring procedures in ST, while this behavior all but disappears (3%) in NST. This suggests, again, that failing to explore is in large part an effort to avoid complexity.

**Result 4** *13% of High Comprehension subjects in the ST treatment are Non-Exploring, but this behavior virtually disappears in the NST treatment.*

More broadly, the decision to use Optimal procedures is dramatically influenced by complexity costs. 35% of ST subjects (and 50% of high comprehension ST subjects) are grouped as following relatively complex 3- and 4-state Optimal procedures, but this rate more than doubles to 72% (85% among the high comprehension) when we remove complexity costs in the NST treatment. This is strong evidence that most subjects have little difficulty understanding how to optimize in this game (the rules and instructions are virtually identical in the two treatments), and that failures to optimize in the ST treatment are driven by complexity costs.

**Result 5** *35% of subjects in the ST are grouped as Optimal in ST, a rate which more than doubles to 72% when complexity costs are removed in NST.*

Third, very few subjects behave consistently with 2-state randomizing procedures. A large number of ST subjects switch from 1A-0 and 1A-I at rates that are both (i) interior and (ii) similar to one another, as required by such procedures. However, virtually all of these subjects take second-arm dominated actions at a high rate and are therefore inconsistent with these procedures, showing evidence instead of using Non-Tracking procedures.

Finally, and crucially, subjects are far more likely to use simplifying hardcoded procedures (indicated by an 'H' between plots in each panel) in ST (50% of high comprehension subjects) than in NST (21% of high comprehension subjects). This pattern is interesting because hardcoded procedures are highly unlikely to arise accidentally and serve little function other than to remove state complexity (recall that they remove one state from both Optimal and Non-Exploration procedures). The fact that they are used somewhat rarely in NST but with high frequency in ST supports Hypothesis 2 and is particularly strong evidence that subjects deliberately shape their procedures to avoid state complexity.

**Result 6** *50% of high comprehension subjects in ST use hardcoded procedures, but less than half as many (21%) do so in NST.*

### 4.3 Classification and State Complexity

We now use the results reported in the previous subsection to identify the state complexity of the procedures subjects choose. To operationalize our classification, we say a subject switches at a "high rate" if she switches at a rate greater than 0.75, at a low rate if she switches at a rate lower than 0.25 and with interior likelihood if she switches at a rate in the inner quartile range of $[0.25, 0.75]$. Following Table 1, we then classify subjects in the following way:

- **1-state (Non-Tracking):** Switches in *each of the four* histories with interior likelihood.
- **1-state (Non-Exploring):** Exhibits hardcoding and explores at a low rate ($p_{1A-I} \leq 0.25$).
- **2-state (Non-Exploring):** Does not exhibit hardcoding and explores at a low rate ($p_{1A-I} \leq 0.25$).
- **2-state (Randomizing):** Switches from the first arm at similar rates regardless of history ($p_{1A-I}$ and $p_{1A-0}$ are within 0.25 of one another), switches from low paying arms at a high rate ($p_{2A-0} \geq 0.75$), switches from high paying arms at a low rate ($p_{2A-1} \leq 0.25$).
- **3-state (Optimal):** Exhibits hardcoding, explores at a high rate ($p_{1A-I} \geq 0.75$), avoids returning to low-payoff arms ($p_{1A-0} \leq 0.25$), switches from low paying arms at a high rate ($p_{2A-0} \geq 0.75$), switches from high paying arms at a low rate ($p_{2A-1} \leq 0.25$).
- **4-state (Optimal):** Does not exhibit hardcoding, but otherwise acts identical to 3-state (Optimal).

Visual inspection of Figure 4 suggests that varying the boundaries we use to define high, low and interior rates would lead to only minor changes to our classification.

At the very bottom of the lower plot in each panel of Figure 4 we list the state-counts of the procedures assigned based on this classification for each subject. Overall, 87% of ST and 90% of NST subjects can be classified as using one of these six procedures – subjects whose behavior is inconsistent with any of these procedures are marked with a dash ('-'). The rough categorization discussed in the preceding subsection (based on the grouping in Figure 4) lines up quite well with this more stringent classification. Almost all subjects grouped as 'Optimal' choose 3- and 4-state procedures and all subjects grouped as Non-Exploring use 1- and 2-state procedures. Most subjects grouped as Non-Tracking use 1-state procedures, though most unclassified subjects are also in this

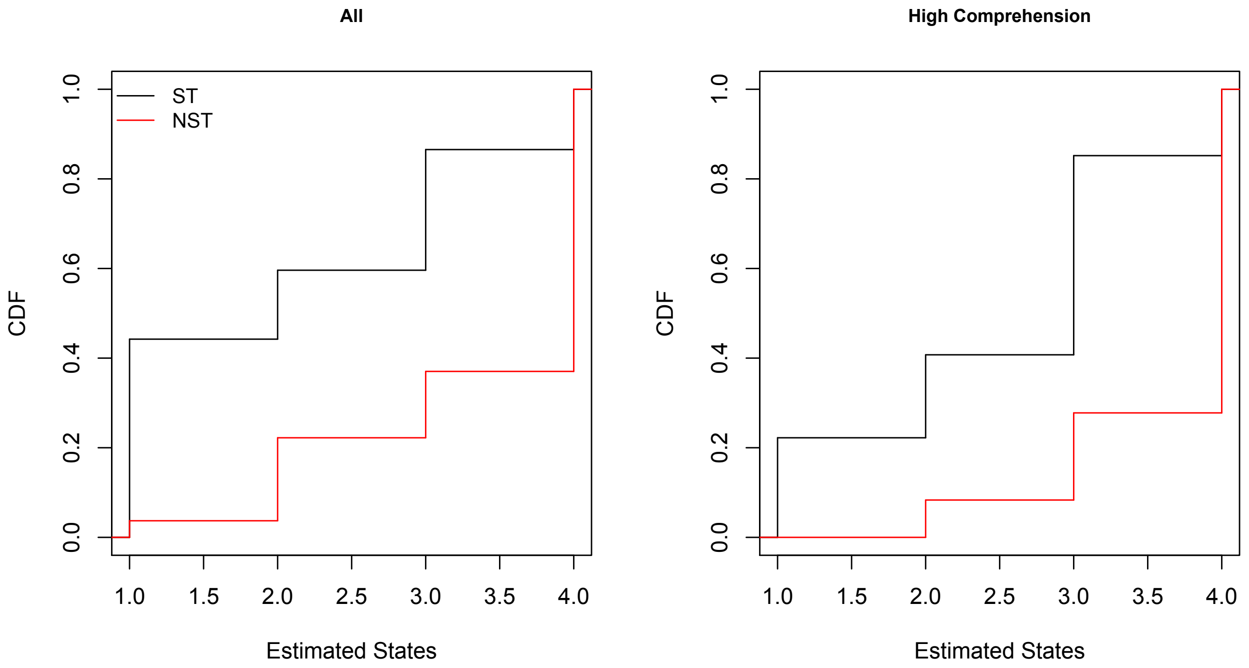


Figure 5: Empirical cumulative density functions of the state counts from classified procedures by treatment. The left panel shows the entire sample, and the right the subset that made no more than one mistake on the comprehension quiz.

group. Most of the subjects not classified as Non-Tracking, Non-Exploring or Optimal in the last section are classified as using 2-state Random procedures, though several of these are untyped.

Figure 5 summarizes classified subjects' procedures by plotting the empirical cumulative density functions of procedural state-counts for each treatment. The left panel shows the results for the entire sample and the right panel for high comprehension subjects. There is a sizable and significant shift in the state complexity of procedures used in ST relative to NST (via Kolmogorov-Smirnov tests), for both high comprehension subjects and overall ($p < 0.001$ in both cases).

**Result 7** *Subjects use significantly lower-state procedures when state complexity costs are high (ST) than when they are low (NST).*

There are two components to this shift. First, as in Hypothesis 1, when subjects must track states themselves (ST), they are far more likely to abandon optimal procedures in favor of suboptimal lower state procedures. Among high comprehension subjects, almost half of ST subjects use suboptimal procedures with fewer than 3-states but fewer than 10% of NST subjects use these lower-state procedures (a significant difference, $p < 0.001$ by a Pearson's proportions test).

**Result 8** *Subjects are more than four times as likely to use simplified suboptimal procedures in ST than in NST*

Second, as in Hypothesis 2, even among subjects that optimize, behavior is systematically different across the two treatments. Most (79%) of high comprehension optimal NST subjects use maximally complex 4-state procedures, with far fewer bothering to hardcode their procedures in order to remove a state from the rule. For ST subjects, these proportions are reversed, with 75% using hardcoded 3-state procedures and only a minority using 4-state procedures ($p < 0.001$ by a Pearson's proportions test).

**Result 9** *Subjects optimize in a systematically different way when exposed to complexity costs. In ST 75% of optimizing subjects use simpler 3-state rules, while in NST 79% use more complex 4-state procedures.*

### 4.4 Procedural Choice vs. Mistakes

Our results show that varying the cost of tracking states leads to a fundamental reduction in the complexity of the procedures subjects implement. Our maintained assumption has been that this is driven by deliberate efforts by subjects to avoid state-complexity. An alternative explanation is that subjects simply make more *mistakes* in ST than in NST because it is more *difficult* to correctly implement optimal procedures in the former than the latter. There are several strong reasons to reject this alternative account of the data and, because this is important for interpreting our findings, we discuss them in some depth.

First, as discussed above, our main results are unlikely to be driven by mistakes stemming from confusion about the game. Instructions for ST and NST are nearly identical and when we condition on subjects that made very few mistakes in comprehension quizzes, our treatment differences remain (and are, in some cases, strengthened). Therefore, if the treatment effects are due to mistakes, they must be due to memory lapses or accidental "slips of the hand" in implementation rather than confusion about the payoff consequences of actions.

Second, most of the simplifying procedures we observe in ST require subjects to take highly *consistent* actions that are unlikely to arise by accident. For instance, in order to use hardcoded 3-state and 1-state procedures, subjects must consistently choose the same initial arm across decision tasks. Subjects are far *more* likely to show this sort of consistency in ST than in NST. Likewise, non-exploring 1- and 2-state procedures require subjects to consistently choose the same arm across periods within a task and, again, this consistency is more common in ST than in NST. These highly patterned behaviors account for most of the simplifying shifts in procedures observed in the ST treatment and it is not plausible that they represent accidental mistakes.

Third, even the random-looking behavior of subjects classified as using Non-Tracking procedures are unlikely to be driven by accidental mistakes (e.g. memory lapses or slips of the hand) made in the pursuit of optimal choice. It is important here to highlight that it is not much more difficult to avoid second-arm dominated actions (the signature deviation made in Non-Tracking procedures) in ST than in NST, if the subject chooses to do so. The only difference between the two cases is that, in the former, the subject must remember her immediately previous period's action. In order for apparent Non-Tracking behavior to be driven by mistakes, these subjects would have to be systematically incapable of remembering immediately previous actions, forcing them to choose arbitrarily instead. This seems highly implausible given what we know about the memory capacities of decision makers; more plausible is that these subjects are perfectly capable of remembering their prior actions but have decided, as part of a broader strategy, to avoid conditioning their behavior on history due to a distaste for complexity.

Fourth, there is recent evidence that decision makers respond to a propensity for error in implementing complex procedures not by persistently making mistakes, but instead by choosing simpler procedures in which mistakes are less likely to occur. Proto et al. (2022) argues that mistakes in implementing an automaton procedure occur due to failures to properly make transitions between states. They provide evidence that lower IQ subjects tend to select simpler strategies in repeated prisoner's dilemmas and argue that this is driven by their higher likelihood of making mistakes when using more complex procedures. In this sense, propensity to make mistakes can be interpreted as another aspect of the cost of complex procedures and another motivation for the selection of simpler procedures.

Finally, we ran an additional diagnostic treatment to study how behavior changes when we make the problem more difficult (easier to make mistakes in) without directly altering the costs of tracking states. Our State Tracking + Distraction (ST+D) treatment replicates the ST treatment but requires subjects to additionally remember a sequence of letters in a separate, paid task, shown between the periods of the bandit problem (details are in Appendix B). While the NST and ST treatments vary the task in a relatively targeted way (aimed specifically at influencing the cost of tracking states), the ST+D treatment simply adds background difficulty to the task by taxing subjects' attention and mental resources. Despite requiring more effort (subjects take nearly twice as long, on average, to make decisions each period in ST+D than they do in ST), ST+D has no effect on the procedures identified in the data relative to ST: as we show in Appendix B, the distribution of procedures is virtually identical in ST and ST+D. The main effect of this increase in task difficulty is instead to add noise, increasing the proportion of "untyped" subjects from 13% to 23% in ST-CL relative to ST. We interpret this as further evidence in favor of the behavioral hypothesis advanced by the automata literature. It is only when we alter a specific feature of a task, linked to underlying automaton characteristics, that we observe a systematic change in the procedures selected by subjects (adding mere difficulty just adds noise).

## 4.5 Complexity Costs vs. Learning

Another competing explanation for our results is that the NST treatment might make optimal procedures not only less costly to implement, but also easier to recognize and therefore easier to learn. The NST treatment, after all, provides subjects with salient reminders of what they have not yet learned during the task, highlighting an "information gap" (e.g. Golman & Loewenstein

(2018)) in payoffs for subjects employing e.g. non-exploration procedures. This salience might help subjects to think more clearly about the payoff advantages of optimal procedures, causing them to learn the optimal procedures at a higher rate during the first half of the experiment in NST. As a result, subjects might use optimal procedures more frequently once they enter the window of our analysis (i.e. the second half of the experiment) in the NST treatment, generating our treatment effect for reasons other than complexity costs.

To examine this possibility, we conducted an additional diagnostic treatment that maintains the learning afforded in the NST treatment, but includes the state-tracking costs of the ST treatment. In our NST-ST treatment, we start subjects in the NST treatment and have them play under that condition for ten tasks. After the tenth task we surprise subjects by telling them that the computer will no longer track states for them (i.e. the computer will no longer remind subjects of what arms they've selected or how much those arms have paid). Subjects thus get the same opportunity to learn from the NST display during the first half of the experiment as in the NST treatment, but once they enter the window of our analysis (the second half of the experiment) they suffer the same state-tracking costs as in the ST treatment.[25]

The NST-ST treatment allows us to use within-subject evidence to test for the effects of state complexity costs in a direct way that is difficult to confound with learning. Under the hypothesis that complexity costs drive procedural choice, we should observe NST-ST subjects responding to the introduction of complexity costs in the second half of the session by *switching* to lower-state procedures. The NST treatment serves as a natural benchmark and control: under our complexity hypothesis, we should observe fewer subjects switching to a lower state procedure in NST (where state complexity costs are *not* introduced) than in NST-ST (where they are). This comparison controls for NST-specific learning by explosing subjects to the exact same salient NST feedback in the first part of the experiment, and looks for differential within-subject changes in behavior in the second part. This is a particularly strenuous test of our hypothesis because we should expect (based on prior experimental evidence) for subjects to be prone to hysteresis and reluctant to change their behavior after ten tasks of play under NST – a well-documented tendency that pushes against our finding a complexity effect.[26]

[25] We collected 61 subjects of data in the NST-ST treatment on Prolific in January 2022. Details are given in Appendix B.

[26] Indeed, we should expect this effect to be particularly severe here. A key finding in Oprea (2020) is that complexity costs fall as subjects gain experience using a procedure, meaning we should expect experience using optimal procedures under NST to lower the costs of implementing them in the second half of NST-ST. For this reason, we should *not* expect to see second-half behavior in NST-ST that looks identical to second half ST behavior, even under the hypothesis that state complexity costs are the major driver of the treatment differences. Optimal procedures should instead be significantly less costly in the second half of NST-ST than of ST, leading to somewhat lower use of optimal procedures in the latter than in the former. Using Kolmogorov-Smirnov tests, we find that, indeed, state counts are marginally higher in the last half of NST-ST than in the last half of ST ($p = 0.06$). However, crucially for our purposes, we find that last half state counts are strongly significantly *lower* in NST-ST than in NST ($p < 0.01$).

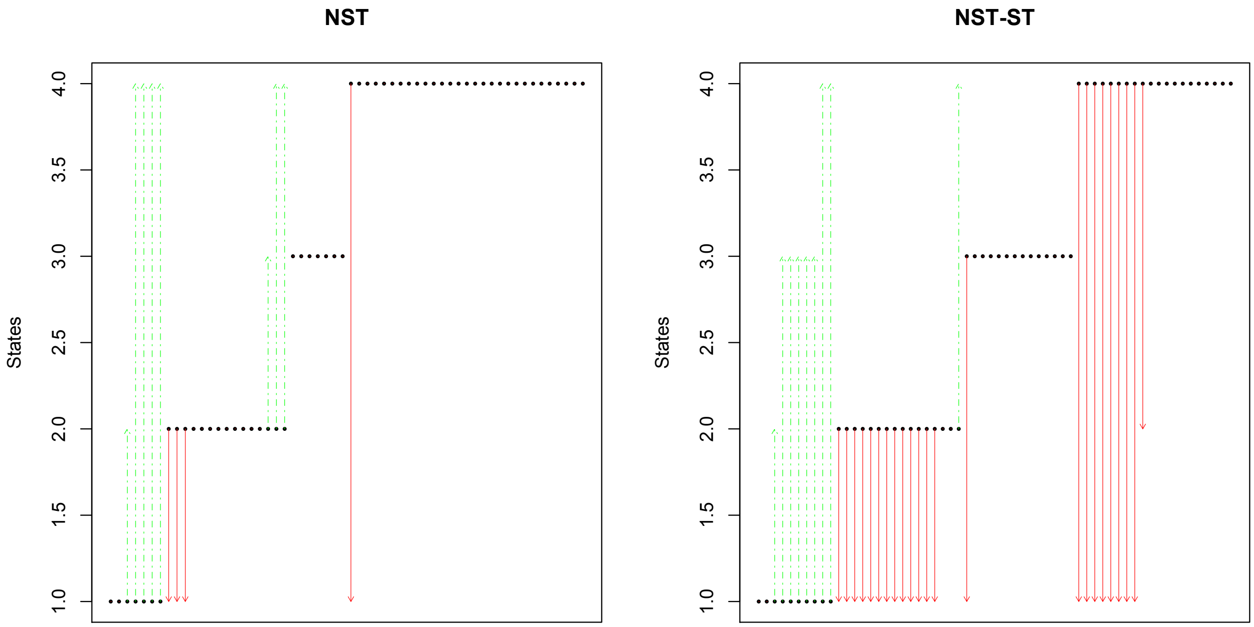


Figure 6: Changes in procedural state counts between the first and second half of the experiment. Dots show estimates of first-half state counts. Arrows show increases (dotted lines) and decreases (solid lines) in state-counts between the first and second half of the experiment. The left panel shows data from the NST treatment and the right panel from the NST-ST treatment.

The right hand panel of Figure 6 plots a dot representing the estimated state count of procedures used during the first ten tasks (i.e. when state-tracking costs are absent) for every subject in the NST-ST treatment. Arrows show the change in procedural state-counts between the first and second half of the experiment. Dashed, upward pointing arrows show *increases* in estimated state counts, while solid downward arrows show corresponding *decreases* in estimated state counts. For comparison, the left hand panel shows the same analysis for the NST treatment in which first half and second half data is measured under the same treatment.

The Figure shows strong evidence in favor of the complexity hypothesis. A minority of subjects in the NST treatment show evidence of learning, with 13% of subjects choosing more complex procedures in the second half relative to the first (upward pointing dashed arrows). Similar learning occurs in the NST-ST treatment (15% of subjects transition to higher state count procedures).[27] However, while virtually no subjects transitions to simpler procedures in the second half of NST (7%), a large fraction (38%) of subjects make this transition in NST-ST. Subjects are thus overall

[27] As it turns out, there is little evidence that this learning is a special outgrowth of the NST treatment. There is also similar learning in the ST treatment between the first and second half, with 9% of subjects switching to more complex procedures.

more than five times more likely to transition to simpler procedures in NST-ST than NST, and unlike NST, NST-ST subjects are far more likely to transition to simpler than to more complex procedures. As a result, second half state counts are significantly lower in NST-ST than NST ($p < 0.01$). This is strong, direct evidence that complexity costs causally lead to the selection of simpler-than-optimal procedures in our tasks.

**Result 10** *A large fraction of subjects in NST-ST transition to less complex procedures after state-tracking costs are introduced. By contrast, almost no subjects reduce the complexity of their procedures over time in the NST treatment.*

## 5 Discussion

How much of human behavior can be explained by the simple idea that people dislike doing complex things? This notion has a long history in economics, but it has received surprisingly little direct empirical attention and, perhaps as a consequence, is rarely invoked to explain empirical puzzles in economics. We conduct a particularly direct empirical test of the organizing power of this idea, framing our investigation using formal automata models of procedural complexity that make predictions amenable to empirical study. Our findings provide strong support for this theory: even in our very basic decision problem, aversion to procedural complexity has a first order effect on behavior. Distaste for procedural complexity may therefore be important for interpreting and predicting behavior in a much wider range of settings.

We study bandit problems simple enough to allow us to identify the procedures most subjects use to guide their decisions. By varying the burden of complexity in a way directly informed by automata theory, we can therefore observe how the choice of procedures responds to complexity. Moreover, by choosing a very targeted intervention – eliminating the effort required to track states without changing the timing, uncertainty or payoffs of the problem – we are able to crisply identify the mechanism by which our intervention alters behavior. We find that subjects use maximally complex 4-state procedures when complexity burdens are artifically removed (i.e., when no cognitive effort is required to track states), but economize on complexity by substituting systematically to simpler, lower-state procedures when normal complexity burdens are present. Our design rules out perennial alternatives like confusion, uncertainty aversion and task-difficulty as explanations. The results therefore strongly support the central hypothesis offered by automata models of complexity: that decision makers systematically choose to implement simple procedures *because* they dislike implementing complex ones.

Our results suggest that, as hypothesized in automata models, there is an "algorithmic layer" lying between primitives like preferences/beliefs and actions that is important for understanding behavior: people have preferences directly over the properties of the procedures guiding their choices and, in particular, dislike implementing procedures that are complex. That is, people have direct preferences not only over the consequences of their choices, but also the process by which these choices are produced because of the way these processes tax cognitive resources. Because of this, human behavior is (at least in part) shaped by features of decision problems that are not reducible to the standard suite of primitives usually studied in economic theory.

Importantly, these preferences have structure and are therefore amenable to modeling, prediction and integration into economic theory. For instance, our work shows that people organize their behavior to avoid rules with many states. Oprea (2020) provides related evidence using a less direct empirical approach: instead of varying complexity costs and looking at its effect on behavior as we do, he directly elicits willingness-to-pay to avoid implementing abstract procedures. Using these methods, Oprea (2020) provides evidence suggesting that people also dislike rules with many transitions or rules that require perpetual transition between states. Mapping the structure of these preferences empirically (using methods like those in Oprea (2020)) and examining their organizing power over real choice in a broader set of economically relevant environments (using methods like ours) is a potentially important empirical enterprise.

Because this sort of algorithmic structure underlies decision making across social and economic life (in all but the simplest settings), the scope of application for this approach seems significant. We highlight two settings within economics in which procedural complexity may be a particularly valuable conceptual and empirical tool. First is the study of repeated strategic interaction, where multiplicity of equilibrium hampers prediction and estimation in theoretical and applied work, respectively. The original aim of automata models was to make repeated game theory more predictive by limiting the set of strategies available to agents in a principled way. Using methods like ours to understand how complexity interacts with other selective forces (e.g. avoidance of strategic risk) may eventually allow for the development of empirically informed predictive theoretical tools, rooted in agents' motivation to economize on complexity. Doing this will require methodological advances that can empirically tease out and separately identify the effects of procedural complexity from alternative forces, such as computational limitations or distaste for strategic risk.

No less promising an application is behavioral economics, where identification of unified explanations for its large and growing list of departures from neoclassical benchmarks seems particularly valuable. How much of behavioral economics can be parsimoniously understood as avoidance of the complex procedures often required to behave optimally? On the surface, the possibility seems promising: many (if not most) departures from neoclassical benchmarks seem to involve simpler-than-optimal behavior. Moreover, automata models have been used to theoretically explain a range of behavioral phenomena including satisficing, primacy and recency effects, choice overload and status quo bias (Salant 2011), stochastic choice (Kalai & Solan 2003), non-Bayesian inference (Chauvin 2020), biases in information processing (Wilson 2014), and failures of backwards induction (Neyman 1985). Use of methods like ours to examine the degree to which patterns in behavioral economics arise due to procedural complexity seems like a particularly promising enterprise for future work.

# References

Abreu, D. & Rubinstein, A. (1988), 'The Structure of Nash Equilibrium in Repeated Games with Finite Automata', *Econometrica* **56**(6), 1259–1281.

Alaoui, L. & Penta, A. (2016), 'Endogenous Depth of Reasoning', *The Review of Economic Studies* **83**(4), 1297–1333.

Anderson, C. M. (2012), 'Ambiguity aversion in multi-armed bandit problems', *Theory and Decision* **72**, 15–33.

Aumann, R. J. (1981), Survey of Repeated Games, *in* 'Essays in Game Theory and Mathematical Economics in Honor of Oskar Morgenstern', Bibliographisches Institut, pp. 11–42.

Banks, J., Olson, M. & Porter, D. (1997), 'An experimental analysis of the bandit problem', *Economic Theory* **10**(1), 55–77.

Banovetz, J. (2020), 'Simple bandits in the lab', *Unpublished Manuscript* .

Börgers, T. & Morales, A. (2004), 'Complexity Constraints in Two-Armed Bandit Problems: An Example'.

Caplin, A., Csaba, D., Leahy, J. & Nov, O. (2019), 'Rational Inattention, Competitive Supply, and Psychometrics'.

Caplin, A., Dean, M. & Martin, D. (2011), 'Search and Satisficing', *American Economic Review* **101**(7), 2899–2922.

Carvalho, L. & Silverman, D. (2019), 'Complexity and Sophistication'.

Cason, T. N. & Mui, V.-L. (2019), 'Individual Versus Group Choices of Repeated Game Strategies: A Strategy Method Approach', *Games and Economic Behavior* **114**, 128–145.

Chauvin, K. P. (2020), 'Euclidean Properties of Bayesian Updating'.

Dal Bó, P. & Fréchette, G. R. (2011), 'The Evolution of Cooperation in Infinitely Repeated Games: Experimental Evidence', *American Economic Review* **101**(1), 411–429.

Dal Bó, P. & Fréchette, G. R. (2019), 'Strategy Choice In The Infinitely Repeated Prisoners' Dilemma', *American Economic Review* p. forthcoming.

Dean, M. & Neligh, N. (2019), 'Experimental Tests of Rational Inattention'.

Dewam, A. & Neligh, N. (2020), 'Estimation Information Cost Functions in Models of Rational Inattention', *Journal of Economic Theory* **187**.

Engle-Warnick, J., McCausland, W. J. & Miller, J. H. (2007), 'The Ghost in the Machine: Inferring Machine-Based Strategies from Observed Behavior'.

Enke, B. & Graeber, T. (2020), ‘Cognitive Uncertainty’.

Frechette, G. & Yuksel, S. (2017), ‘Infinitely repeated games in the laboratory: Four perspectives on discounting and random termination’, *Experimental Economics* **20**(2), 279–308.

Gabaix, X., Laibson, D., Moloche, G. & Weinberg, S. (2006), ‘Costly Information Acquisition: Experimental Analysis of a Boundedly Rational Model’, *American Economic Review* **96**(4), 1043–1068.

Gale, D. & Sabourian, H. (2005), ‘Complexity and Competition’, *Econometrica* **73**(3), 739–769.

Gans, N., Knox, G. & Croson, R. (2007), ‘Simple models of discrete choice and their performance in bandit experiments’, *Manufacturing & Service Operations Management* **9**(4), 383–408.

Golman, R. & Loewenstein, G. (2018), ‘Information Gaps: A Theory of Preferences Regarding the Presence and Absence of Information’, *Decision* **5**, 143–164.

Halevy, Y. & Mayraz, G. (2020), ‘Modes of Rationality: Act versus Rule-Based Decisions’.

Hoelzemann, J. & Klein, N. (2019), ‘Bandits in the lab’, *Unpublished Manuscript* .

Hopcroft, J. E. & Ullman, J. D. (1979), *Introduction to Automata Theory: Languages and Computation*, Addison-Wesley.

Hudja, S. & Woods, D. (2019), ‘Behavioral bandits: Analyzing the exploration versus exploitation trade-off in the lab’, *Unpublished Manuscript* .

Jones, M. T. (2014), ‘Strategic Complexity and Cooperation: An Experimental Study’, *Journal of Economic Behavior & Organization* **106**, 352–366.

Kalai, E. & Solan, E. (2003), ‘Randomization and Simplification in Dynamic Decision-Making’, *Journal of Economic Theory* **111**(2), 251–264.

Kalai, E. & Stanford, W. (1988), ‘Finite Rationality and Interpersonal Complexity in Repeated Games’, *Econometrica* **56**(2), 397–410.

Kool, W. & Botvinick, M. (2018), ‘Mental Labour’, *Nature Human Behavior* **2**, 899–908.

Meyer, R. J. & Shi, Y. (1995), ‘Sequential choice under ambiguity: Intuitive solutions to the armed-bandit problem’, *Management Science* **41**(5), 817–834.

Neyman, A. (1985), ‘Bounded Complexity Justifies Cooperation in the Finitely Repeated Prisoners’ Dilemma’, *Economics Letters* **19**(3), 227–229.

Nielsen, K. & Rehbeck, J. (2020), ‘When Choices are Mistakes’.

Oprea, R. D. (2020), ‘What Makes a Rule Complex’, *American Economic Review* **110**(12), 3913–3951.

Proto, E., Rustichini, A. & Sofianos, A. (2022), ‘Intelligence, Errors, and Cooperation in Repeated Interactions’, *Review of Economic Studies* p. forthcoming.

Romero, J. & Rosokha, Y. (2018), ‘Constructing Strategies in the Indefinitely Repeated Prisoner’s Dilemma Game’, *European Economic Review* **104**, 185–219.

Romero, J. & Rosokha, Y. (2019), ‘The Evolution of Cooperation: The Role of Costly Strategy Adjustments’, *American Economic Journal: Microeconomics* **11**(1), 299–328.

Rubinstein, A. (1986), ‘Finite Automata Play the Repeated Prisoner’s Dilemma’, *Journal of Economic Theory* **39**(1), 83–96.

Sabourian, H. (2004), ‘Bargaining and Markets: Complexity and the Competitive Outcome’, *Journal of Economic Theory* **116**(2), 189–228.

Salant, Y. (2011), ‘Procedural Analysis of Choice Rules with Applications to Bounded Rationality’, *American Economic Review* **101**(2), 724–748.

Sanjurjo, A. (2017), ‘Search With Multiple Attributes: Theory and Empirics’, *Games and Economic Behavior* **104**, 535–562.

Shenhav, A., Musslick, S., Lieder, F., Kool, W., Griffiths, T. L., Cohen, J. D. & Botvinick, M. M. (2017), ‘Toward a Rational and Mechanistic Account of Mental Effort’, *Annual Review of Neuroscience* **40**, 99–124.

Simon, H. A. (1955), ‘A Behavioral Model of Rational Choice’, *The Quarterly Journal of Economics* **69**(1), 99–118.

Wilson, A. (2014), ‘Bounded Memory and Biases in Information Processing’, *Econometrica* **82**(6), 2257–2294.

# Online Appendices

# A Theoretical Benchmarks

In this Appendix, we derive two benchmarks of interest for our experiment. In Appendix A.1, we derive the critical value for exploration in our bandit problem. In Appendix A.2 we derive the optimal random transition probability for the 2-state partially randomized procedure in Section 2. These derivations closely follow Börgers & Morales (2004) and we refer the reader there for further analysis.

## A.1 Critical value for exploration

We can find a critical value such that a decision maker would never explore. Intuitively, if $x$ is close to 1, the cost of exploration (the risk of earning a zero) outweighs benefit (the chance of earning one). With discount rate $\delta$, the condition is:

$$\left(\frac{x}{1-\delta}\right) \geq \frac{1}{3}\left(\frac{\delta x}{1-\delta}+\frac{x}{1-\delta}+\frac{1}{1-\delta}\right) \tag{1}$$

Setting these two equal, we can solve for the critical value.

$$\overline{x} = \frac{1}{1+(1-\delta)} \tag{2}$$

For values larger than $\overline{x}$, it is optimal to remain on the initial arm and never explore.

## A.2 Optimal Transition Probability for Partially Random 2-State Procedure

Consider first the value function when the arms are $(x, 0)$ and the DM's strategy dictates a pull of the initial arm:

$$V_0 = x + \delta\big[(1-p)V_0 + p\delta V_0\big] \tag{3}$$

The DM's first choice yields a payoff of $x$, and we discount future payoffs by $\delta$. In the next period, with probability $(1-p)$ she pulls the initial arm again (leaving her state unchanged). With probability $p$ she pulls the other arm, which pays zero; she then returns to the initial arm. Solving for $V_0$:

$$V_0 = \frac{x}{(1-\delta)(1+\delta p)} \tag{4}$$

Next, consider the value function when the arms are $(x, x)$:

$$V_x = \frac{x}{1-\delta} \tag{5}$$

If both arms pay $x$, then the DM's pattern of choices do not affect payoffs. Finally, consider the value function when the arms are $(x, 1)$ and the DM's strategy dictates a pull of the initial arm:

$$V_1 = x + \delta\left[(1-p)V_1 + p\left(\frac{1}{1-\delta}\right)\right] \tag{6}$$

The DM pulls the initial arm, earning $x$. She then stays on the initial arm with probability $(1-p)$; with probability $p$, she pulls the other arm and stays forever. Solving for $V_1$:

$$V_1 = \frac{(1-\delta)x + \delta p}{(1-\delta)(1-\delta+\delta p)} \tag{7}$$

In the optimal 2-State strategy, when the DM plays the initial arm, she cannot remember if she has tried the other arm (or what it pays). When her strategy dictates a pull of the initial arm, her value function is the average of $V_0$, $V_x$, and $V_1$.

$$V = \frac{1}{3}\left[\frac{x}{(1-\delta)(1+\delta p)} + \frac{x}{1-\delta} + \frac{(1-\delta)x+\delta p}{(1-\delta)(1-\delta+\delta p)}\right] \tag{8}$$

We can maximize this function with respect to $p$ to find the optimal experimentation probability. Note that maximizing $V$ with respect to $p$ is the same as maximizing:

$$M = \frac{x}{1+\delta p} + \frac{(1-\delta)x+\delta p}{(1-\delta+\delta p)} \tag{9}$$

To find the maximum, we consider the first and second derivatives.

$$\frac{dM}{dp} = -\frac{\delta x}{(1+\delta p)^2} + \frac{\delta(1-\delta)(1-x)}{(1-\delta+\delta p)^2} \tag{10}$$

$$\frac{d^2M}{dp^2} = \frac{2\delta^2 x}{(1+\delta p)^3} - \frac{2\delta^2(1-\delta)(1-x)}{(1-\delta+\delta p)^3} \tag{11}$$

The second derivative is negative when:

$$p < \frac{(1-\delta)^{\frac{1}{3}}(1-x)^{\frac{1}{3}} - (1-\delta)x^{\frac{1}{3}}}{\delta(x^{\frac{1}{3}} - (1-\delta)^{\frac{1}{3}}(1-x)^{\frac{1}{3}})} \tag{12}$$

Recall that $p$ must be greater than zero. Consider the numerator and the denominator separately. The numerator is positive when:

$$(1-\delta)x^{\frac{1}{3}} < (1-\delta)^{\frac{1}{3}}(1-x)^{\frac{1}{3}} \tag{13}$$

$$x < \frac{1}{1+(1-\delta)^2} \tag{14}$$

Note that his value is strictly greater than the critical value in (2). As a result, the numerator will be positive for all values of $x$ such that an DM would consider exploration. The denominator is positive when:

$$x^{\frac{1}{3}} > (1-\delta)^{\frac{1}{3}}(1-x)^{\frac{1}{3}} \tag{15}$$

We can set these equal to solve for a second critical value.

$$\underline{x} = \frac{1-\delta}{2-\delta} \tag{16}$$

For values $x \geq \overline{x}$, we know that $p = 0$. For values $\underline{x} < x < \overline{x}$, we can solve for the optimal $p$ by setting the first derivative in 10 equal to zero and solving for $p$:

$$p = \frac{\sqrt{1-\delta}\sqrt{1-x} - (1-\delta)\sqrt{x}}{\delta\sqrt{x} - \delta\sqrt{1-\delta}\sqrt{1-x}} \tag{17}$$

What about values of $x$ less than the critical value in (16)? Consider a case where $p = 1$, but the first derivative is still positive (i.e., the maximum of $M$ is at the upper boundary of $p$). In this situation, the following inequality holds:

$$\frac{x}{(1+\delta)^2} < \frac{\delta(1-\delta)(1-x)}{(1-\delta+\delta)^2} \tag{18}$$

$$x < \frac{1-\delta}{\frac{1}{(1+\delta)^2} + 1 - \delta} \tag{19}$$

The value in (19) is strictly greater than $\underline{x}$. Thus, for all values of $x$ less than the critical value in (16), the optimal experimentation probability is $p = 1$. Note, however, that (19) also established that there is a range of $x$ such that $\underline{x} < x < \overline{x}$ and $p = 1$, i.e., $p$ as define in (17) may be greater than 1. Thus, the optimal experimentation probability must be defined piece-wise:

$$p = \begin{cases} 0 & \text{if } x \geq \overline{x} \\ \min\left\{\frac{\sqrt{1-\delta}\sqrt{1-x}-(1-\delta)\sqrt{x}}{\delta\sqrt{x}-\delta\sqrt{1-\delta}\sqrt{1-x}}, 1\right\} & \text{if } \underline{x} < x < \overline{x} \\ 1 & \text{if } x \leq \underline{x} \end{cases} \tag{20}$$

# B Diagnostic Treatments

The experiment includes two diagnostic treatments, designed to better understand the mechanism driving our primary ST/NST treatment comparison.

**State Tracking + Distraction (ST-D)**: In this treatment, subjects are assigned the ST treatment, but between each period the software flashes a random letter on subjects' screens. At the end of each five period block, subjects are asked to type these five letters as part of their payment. In addition to per period payoffs of 0, 65 or 100 points in the bandit task, subjects receive 300 payment points every time they correctly type the five-letter sequence. Instead of being paid for each point in excess of 700 (as in the other treatments), they are paid for each point in excess of 1000.[28] The treatment was run using 60 Prolific subjects in April and May 2020.

Estimated switching probabilities from the second half of the sessio are plotted in graphs analogous to Figure 4 in the top half of Figure 7. There are two main findings from this treatment. First is that subjects' procedures are considerably more likely to be impossible to identify in ST+D than in ST (23% rather than 13%), suggesting a higher rate of noisy mistakes due to the increased task difficulty of the treatment. Second is that conditional on being typed, procedural choice is virtually identical in ST-D and ST. Subjects choose 1/2/3/4 state rules 44%/15%/27%/14% of the time in ST and 48%/13%/26%/13% of the time in ST-D. Unsurprisingly we cannot reject the hypothesis that these distributions are identical via a Wilcoxon test ($p = 0.8$). The results thus suggest that

[28] Also, because these sessions are longer, subjects are paid a larger base pay of \$5. rather than the \$2.50 from other treatments.

changes to task difficult lead to noisier behavior rather than systematic changes in the procedures subjects use.

**No State Tracking - State Tracking (NST-ST)**: In this treatment subjects are given the exact same instructions used in the NST treatment and are assigned that treatment for the first ten (of twenty total) tasks. After task ten, subjects are given new instructions informing them that they will no longer be shown question marks under arms they haven't yet selected or payoff amounts under arms they have. That is, they are told they will play the rest of the tasks under the ST treatment. This change of treatment is a surprise to subjects (it is not discussed in the earlier instructions). Payments and incentives are exactly as in the NST and ST treatments. The treatment was run using 61 Prolific subjects in January 2022. Estimated switching probabilities from the second half of the session are plotted in graphs analogous to Figure 4 in the top half of Figure 7. The main findings from this treatment are discussed in detail in Section 4.5.

# C Instructions to Subjects

In this Appendix we reproduce the instructions to subjects. These were deployed in HTML in the experiment and unfolded progressively interspersed with comprehension quiz questions. In Appendix C.1 we reproduce instructions from our ST treatment and in Appendix C.2 instructions from our NST treatment.

## C.1 State Tracking Treatment

1. **Introduction**

    - We will start by providing you with **INSTRUCTIONS** for the study.
    - We will ask you **QUESTIONS** to check that you understand the instructions. You should be able to answer all of these questions correctly.
    - Please read and follow the instructions closely and carefully.
    - If you **COMPLETE** the main parts of the study, you will receive a **GUARANTEED PAYMENT** of **$2.50**.
    - In addition, your **CHOICES** in the GAME portion of the study will result in **PERFORMANCE-BASED EARNINGS**. You will play in **TWENTY (20) GAMES** worth **REAL MONEY**. Your **AVERAGE** points from **ALL TWENTY GAMES** will be converted into an additional payment.
    - After you finish the instructions, you will have a chance to play several **PRACTICE GAMES** before you play for real money.

2. **Two Options to Choose Between**

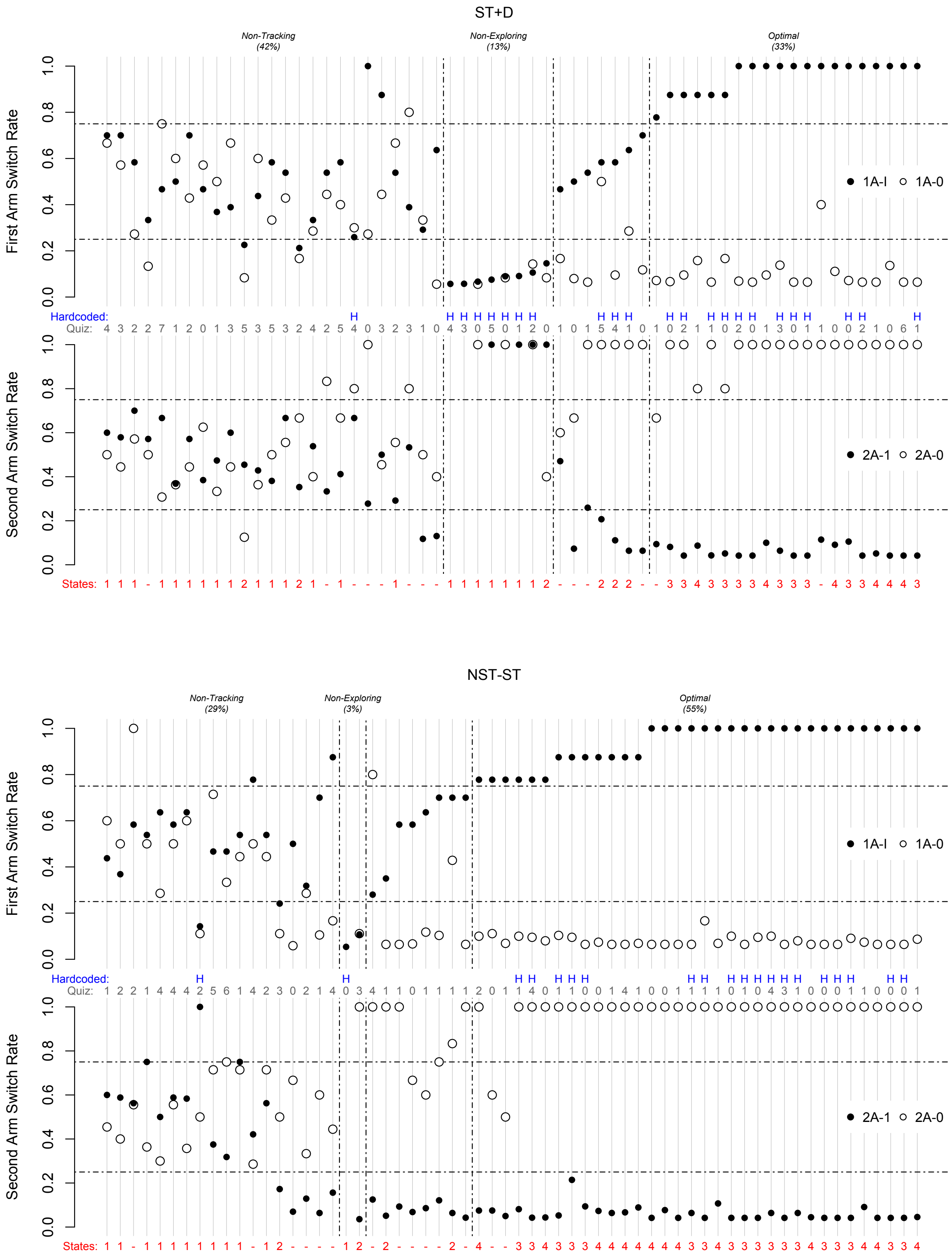
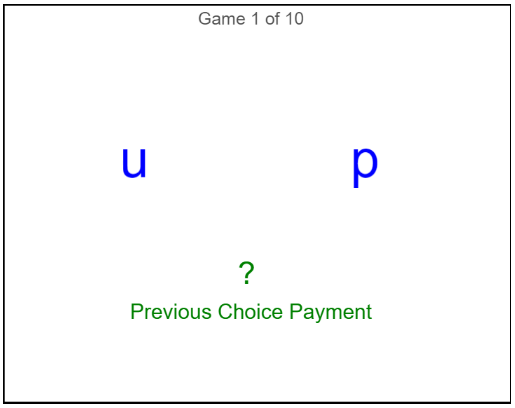
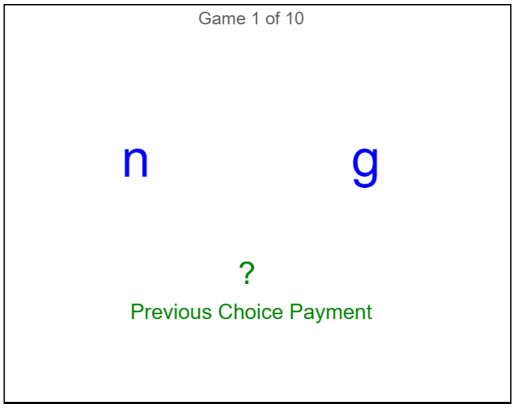


Figure 7: Switching probabilities for each subject in our diagnostic treatments, ST+D (upper panel) and NST-CL (lower panel). *Notes: The upper plot of each panel shows switching behavior after play of the initial arm, the lower plot after play of the second arm. Numbers at the bottom of each panel shows the estimated number of states in the procedure the subject used. Numbers between plots in each panel show the number of of errors the subject made in a comprehension quiz prior to the experiment, and the letter 'H' between plots in each panel designates a subject who made consistent ("hardcoded") initial choices.*



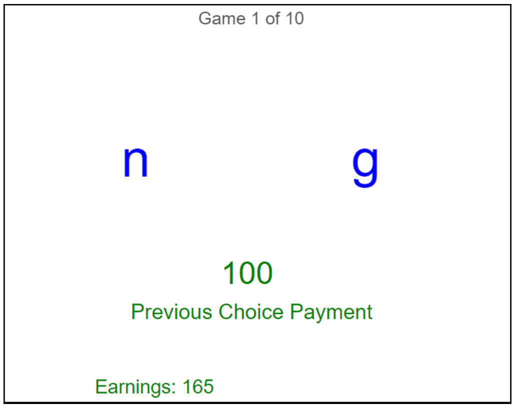
- The experiment is divided into twenty **GAMES**, each of which is is divided into several **CHOICES**.
- In each game, you will repeatedly choose between **TWO OPTIONS** that we will call **EARLIER-LETTER** and **LATER-LETTER**, represented by two letters on your screen.
  - In the example above, the Earlier-Letter option is represented by '**p**' and the later-letter by '**u**' (because '**p**' comes **earlier** than '**u**' in the alphabet).
- Your **FIRST** choice in a game will **ALWAYS PAY 65** points. This is true whether you choose Earlier-Letter or Later-Letter first.
  - For example, suppose your **first** choice were **Earlier-Letter**. Then you would know that Earlier-Letter would pay **65 points every time** you choose it for the rest of the game.
- After your first choice, the **OTHER OPTION** will pay a **VALUE** of either **0**, **65**, or **100**. This value is **INDEPENDENTLY** and **RANDOMLY** determined by the computer before the game begins. Each value (**65**, or **100**) is **EQUALLY LIKELY** to be selected. It remains the **SAME WITHIN A GAME**.
  - For example, suppose your **first** choice were **Later-Letter**. Then **Earlier-Letter** would be equally likely to pay (**0**,**65**, or **100**). If you choose Earlier-Letter and it pays **100**, then you would know that Earlier-letter would pay **100 points every time** you choose it for the rest of the game.
- However, the value of each option **CHANGES BETWEEN GAMES**: once a game ends, payments reset. Your first choice (either Earlier-Letter or Later-Latter) will pay 65, and the other option will get a new random value.

3. **Typing Letters to Make Choices**



- For each **CHOICE**, each of the **TWO OPTIONS** will require you to **TYPE A LETTER**.
  - In the example above, you would need to type '**n**' (lower case 'N') or '**g**' (lower case 'G') to make a choice.
- These **LETTERS** will **RANDOMLY CHANGE** ('a' to 'z') for each choice and be shown in a **RANDOM ORDER** (left or right) on your screen.
- The **EARLIER LETTER** in the alphabet will always represent the Earlier-Letter option; the **LATER LETTER** in the alphabet will always represent the Later-Letter option.
  - In the example above, typing '**g**' will select the Earlier-Letter option (and give you the Earlier-Letter payment) while typing '**n**' will select the Later-Letter option (and give you the Later-Letter payment).

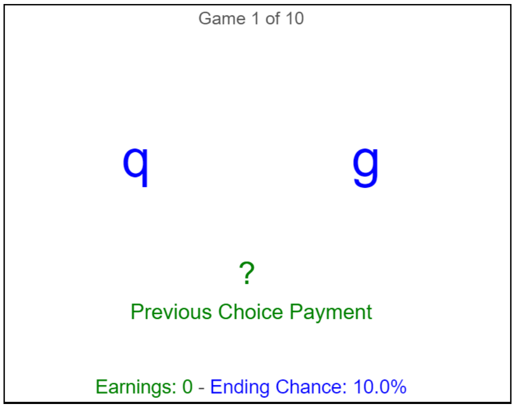
### 4. Tracking Payments



- After you choose an option, the **PAYMENT** you earned (**0**, **65**, or **100**) for that choice appears in the middle of the screen in green. This value always represents your **PREVIOUS CHOICE'S** payment.
  - In the example above, the previous choice paid **100** points.

- Your **EARNINGS** are cumulative for **ALL YOUR CHOICES** so far in the game and appear at the bottom in green.
  - This number is the **sum** of all your payments so far in the game.
  - In the example above, the choices have paid a total of **165** points so far.

5. **Blocks of Choices**



Game 1 of 10

q g

?

Previous Choice Payment

Earnings: 0 - Ending Chance: 10.0%

- The **NUMBER OF CHOICES** you're allowed to make in any game is **RANDOM**. Every time you make a choice, there is a **10% CHANCE** that the computer will make it the **LAST** (paying) choice of the game.
  - A 10% chance that the game ends each choice means that on **average** there will be **10 choices** in the game.
  - Many games will be **shorter**, but others will be **much longer**.
  - The probability each choice is the last **does not depend** on how many choices you have already made. Every choice is equally likely to be the last one that counts.
- In each game, you will always make your choices in **BLOCKS OF FIVE**.
- After every block of five the computer will tell you whether the game actually **RANDOMLY ENDED** during that block. If the computer randomly ended the game during the block (before the last choice of the block), any choices you made **AFTER THE LAST** choice in the block **WON'T COUNT** for payment.
  - Example: If you make **five choices** in a block and the computer randomly **ended** the game on the **third choice** of the block, choices **1, 2 and 3 in the block will count** for payment and choices **4 and 5 in the block will not count** for payment.
- When you have made the **FINAL CHOICE** in a game, the computer will inform you that this has happened and you will start a **NEW GAME**. When a new game starts, the **VALUES WILL CHANGE** for the Earlier-Letter and Later-Letter options. There is no connection between games – each game will be brand new.

6. **Other Instructions**

- You will play in two practice games to familiarize yourself with the software before you play games for real money.
- Because part of the experiment tests your memory, please do not use external tools (e.g., pencil and paper) to assist during the experiment.
- Please do not unnecessarily refresh your browser, as doing so can make the software unstable. Qualtrics tracks your refreshes–excessive refreshes will void your bonus payment.

7. **Cash Payments**

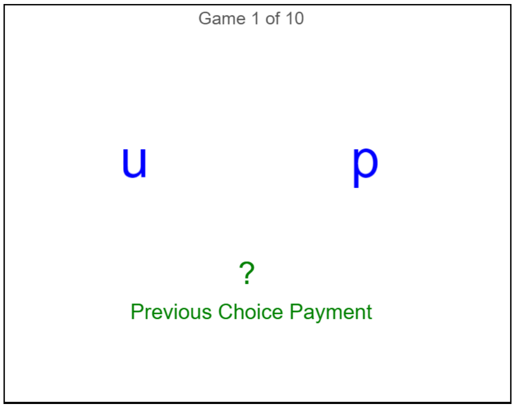
   - You will be paid $2.50 for finishing the experiment. If you decide to leave before finishing, you will forfeit this amount.
   - In addition, you will potentially earn a **PERFORMANCE-BASED BONUS**.
   - Your **POINTS** from **ALL TWENTY (20) GAMES** will be **AVERAGED**.
   - If your **AVERAGE** point total is **GREATER THAN** 700, you will earn a **BONUS**.
     - For every point you earn (on average) greater than 700, you will be paid $0.03 (three cents)
     - For example, if you average **950** points, your bonus would be: (**950** - **700**) * $0.03 = $7.50
     - For example, if you average **600** points, you would not earn a bonus.

## C.2 No State Tracking Treatment

1. **Introduction**

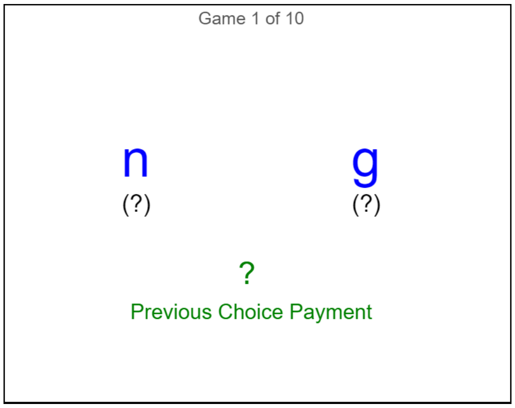
   - We will start by providing you with **INSTRUCTIONS** for the study.
   - We will ask you **QUESTIONS** to check that you understand the instructions. You should be able to answer all of these questions correctly.
   - Please read and follow the instructions closely and carefully.
   - If you **COMPLETE** the main parts of the study, you will receive a **GUARANTEED PAYMENT** of **$2.50**.
   - In addition, your **CHOICES** in the GAME portion of the study will result in **PERFORMANCE-BASED EARNINGS**. You will play in **TWENTY (20) GAMES** worth **REAL MONEY**. Your **AVERAGE** points from **ALL TWENTY GAMES** will be converted into an additional payment.
   - After you finish the instructions, you will have a chance to play several **PRACTICE GAMES** before you play for real money.

2. **Two Options to Choose Between**



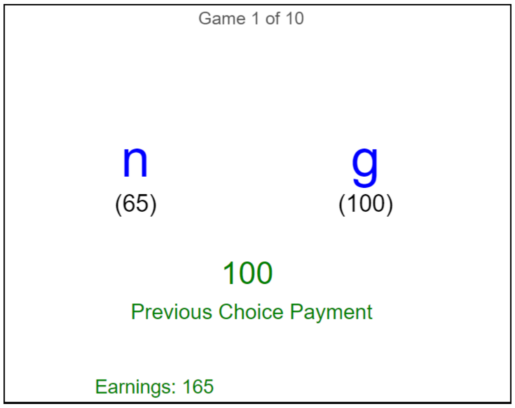
- The experiment is divided into twenty **GAMES**, each of which is is divided into several **CHOICES**.
- In each game, you will repeatedly choose between **TWO OPTIONS** that we will call **EARLIER-LETTER** and **LATER-LETTER**, represented by two letters on your screen.
  - In the example above, the Earlier-Letter option is represented by '**p**' and the later-letter by '**u**' (because '**p**' comes **earlier** than '**u**' in the alphabet).
- Your **FIRST** choice in a game will **ALWAYS PAY 65** points. This is true whether you choose Earlier-Letter or Later-Letter first.
  - For example, suppose your **first** choice were **Earlier-Letter**. Then you would know that Earlier-Letter would pay **65 points every time** you choose it for the rest of the game.
- After your first choice, the **OTHER OPTION** will pay a **VALUE** of either **0**, **65**, or **100**. This value is **INDEPENDENTLY** and **RANDOMLY** determined by the computer before the game begins. Each value (**65**, or **100**) is **EQUALLY LIKELY** to be selected. It remains the **SAME WITHIN A GAME**.
  - For example, suppose your **first** choice were **Later-Letter**. Then **Earlier-Letter** would be equally likely to pay (**0**,**65**, or **100**). If you choose Earlier-Letter and it pays **100**, then you would know that Earlier-letter would pay **100 points every time** you choose it for the rest of the game.
- However, the value of each option **CHANGES BETWEEN GAMES**: once a game ends, payments reset. Your first choice (either Earlier-Letter or Later-Latter) will pay 65, and the other option will get a new random value.

3. **Typing Letters to Make Choices**



- For each **CHOICE**, each of the **TWO OPTIONS** will require you to **TYPE A LETTER**.
    - In the example above, you would need to type ‘**n**’ (lower case ‘N’) or ‘**g**’ (lower case ‘G’) to make a choice.
- These **LETTERS** will **RANDOMLY CHANGE** (‘a’ to ‘z’) for each choice and be shown in a **RANDOM ORDER** (left or right) on your screen.
- The **EARLIER LETTER** in the alphabet will always represent the Earlier-Letter option; the **LATER LETTER** in the alphabet will always represent the Later-Letter option.
    - In the example above, typing ‘**g**’ will select the Earlier-Letter option (and give you the Earlier-Letter payment) while typing ‘**n**’ will select the Later-Letter option (and give you the Later-Letter payment).
- The **QUESTION MARKS** (in parentheses) under each letter indicates that you have not yet tried that option.
    - In the example above, the ‘**(?)**’ under ‘**g**’ indicates Earlier-Letter has not been tried
    - In the example above, the ‘**(?)**’ under ‘**n**’ indicates Later-Letter has not been tried.

4. **Tracking Payments**



Game 1 of 10

n
(65)

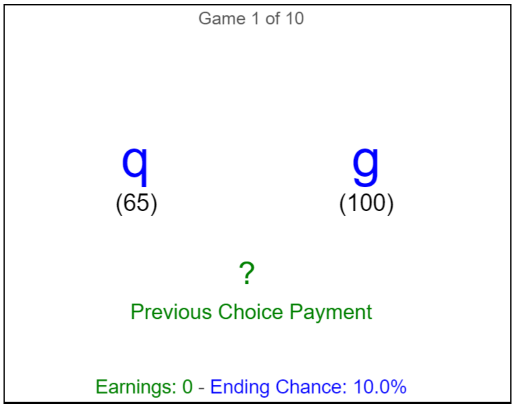
g
(100)

100
Previous Choice Payment

Earnings: 165

- After you choose an option, the **PAYMENT** you earned (**0**, **65**, or **100**) for that choice appears in the middle of the screen in green. This value always represents your **PREVIOUS CHOICE'S** payment.
  - In the example above, the previous choice paid **100** points.
- Your **EARNINGS** are cumulative for **ALL YOUR CHOICES** so far in the game and appear at the bottom in green.
  - This number is the **sum** of all your payments so far in the game.
  - In the example above, the choices have paid a total of **165** points so far.
- After you have **CHOSEN** an option, the **AMOUNT** it pays appears **BELOW**. This number will remain until the game ends and a new one begins
  - In the example above, Earlier-Letter was tried and paid **65**
  - In the example above, Later-Letter was tried and paid **100**

5. **Blocks of Choices**



Game 1 of 10

q
(65)

g
(100)

?
Previous Choice Payment

Earnings: 0 - Ending Chance: 10.0%

- The **NUMBER OF CHOICES** you're allowed to make in any game is **RANDOM**. Every time you make a choice, there is a **10% CHANCE** that the computer will make it the **LAST** (paying) choice of the game.

  - A 10% chance that the game ends each choice means that on **average** there will be **10 choices** in the game.
  - Many games will be **shorter**, but others will be **much longer**.
  - The probability each choice is the last **does not depend** on how many choices you have already made. Every choice is equally likely to be the last one that counts.
- In each game, you will always make your choices in **BLOCKS OF FIVE**.
- After every block of five the computer will tell you whether the game actually **RANDOMLY ENDED** during that block. If the computer randomly ended the game during the block (before the last choice of the block), any choices you made **AFTER THE LAST** choice in the block **WON'T COUNT** for payment.
  - Example: If you make **five choices** in a block and the computer randomly **ended** the game on the **third choice** of the block, choices **1, 2 and 3 in the block will count** for payment and choices **4 and 5 in the block will not count** for payment.
- When you have made the **FINAL CHOICE** in a game, the computer will inform you that this has happened and you will start a **NEW GAME**. When a new game starts, the **VALUES WILL CHANGE** for the Earlier-Letter and Later-Letter options. There is no connection between games – each game will be brand new.

6. **Other Instructions**

- You will play in two practice games to familiarize yourself with the software before you play games for real money.
- Please do not unnecessarily refresh your browser, as doing so can make the software unstable. Qualtrics tracks your refreshes–excessive refreshes will void your bonus payment.

7. **Cash Payments**

- You will be paid $2.50 for finishing the experiment. If you decide to leave before finishing, you will forfeit this amount.
- In addition, you will potentially earn a **PERFORMANCE-BASED BONUS**.
- Your **POINTS** from **ALL TWENTY (20) GAMES** will be **AVERAGED**.
- If your **AVERAGE** point total is **GREATER THAN 700**, you will earn a **BONUS**.
  - For every point you earn (on average) greater than 700, you will be paid $0.03 (three cents)
  - For example, if you average **950** points, your bonus would be: (**950** - **700**) * $0.03 = $7.50
  - For example, if you average **600** points, you would not earn a bonus.