

Generating Random Sequences For You: Modeling Subjective Randomness in Competitive Games

Arianna Yuan (xfyuan@stanford.edu)

Department of Psychology, Stanford University, 450 Serra Mall
Stanford, CA 94305

Abstract

Rapoport and Budescu (1992) showed that despite subjects' failure to generate random sequences under explicit instructions, they were able to generate more random sequences when engaging in competitive games like Matching Pennies. Why people were able to correct their distorted sense of randomness in competitive games remains unclear. Therefore, I explored two probabilistic models to answer this question. The first one is the Coin-Weight Model, which assumes that subjects predict their competitor's choices by implicitly assuming that their competitors intended to generate binary sequences that simulated the outcome of tossing an unbiased coin. The second one is the Markov model, which assumes that subjects believed that their competitors intended to generate sequences that simulated the outcome of a generating process with transition probability equal to 0.5. I found that both the Coin-Weight Model and Markov Model are able to characterize the calibrated subjective randomness in Dyad condition (playing Matching Pennies), but the Markov Model is better than Coin-Weight Model in explaining the Single condition, in which subjects were paired to play Matching Pennies but needed to specify their choice sequences in advance. The current study suggests that the calibrated subjective randomness in competitive games can be simulated by probabilistic models that combine the on-line evaluation of sequence randomness and Theory-of-Mind reasoning.

Keywords: subjective randomness; matching pennies; probabilistic models

Introduction

Previous studies showed that people are relatively poor at generating random sequences (Bakan, 1960). However, people are able to generate more random sequences when feedbacks are available (Neuringer, 1986), like in competitive games (Rapoport & Budescu, 1992). Many theories have been proposed to account for subjects' failure to generate random sequences under explicit instructions (Griffiths & Tenenbaum, 2003), but few of them explain why feedbacks in competitive games help people generate more random sequences. In the current study, I will extend a previous computational model developed by Griffiths and Tenenbaum and also explore a possible Markov Model to provide a plausible explanation for the calibrated subjective randomness in competitive games.

The specific phenomenon I am interested in is people's behavior in a competitive game called matching pennies. In this game, two agents A and B each makes a binary choice (e.g., 0 or 1) secretly. A ("matcher") wins if their choices are the same and B ("non-matcher") wins if their choices are different. Mathematically, it can be proved that the optimal strategy is to choose the two alternatives with equal probability. Rapoport and Budescu found that people are

much better in generating random sequences when involving in such competitive game.

There were 3 conditions in their experiment: Dyad Condition, in which individual subjects were paired to form dyads and each of them played 150 rounds of the matching pennies game; Single Condition, in which everything is the same as in condition D except that the paired dyads were asked to specify their choices in advance for the 150 rounds and were told that the responses would be matched on a trial-by-trial basis to determine the winner; Randomization Condition, in which subjects participated individually and were instructed to generate a sequence of 150 random binary responses to simulate the outcome of tossing an unbiased coin.

The key results I want to model are the patterns of sequential dependencies, i.e., the distributions of successive k -tuples ($k = 2, 3, 4$) were not uniform as expected under a "true" random generating process. The authors calculated the frequencies of k -tuples (e.g., 3-tuple [0 1 1], 4-tuple [0 0 0 0]), and found that in Randomization Condition, subjects were more likely to generate [0 1 0 1] and [1 0 1 0] than [0 0 0 0] and [1 1 1 1]. Therefore, they used two statistics to indicate the extent to which the distributions deviate from the outcome of a random generating process: the mean absolute deviation (MAD) from expectation and the standard deviation of the observed proportions around their expectation (SD). They are defined as:

$$MAD = \sum_{j=1}^{2^k} |p_j - 1/2^k| / 2^k, k = 2, 3, 4.$$

$$SD = \sum_{j=1}^{2^k} [(p_j - 1/2^k)^2 / (2^k - 1)]^{1/2}, k = 2, 3, 4.$$

Here p_j stands for the probability of individual k -tuples.

They found that MAD and SD in Randomization Condition were the biggest, followed by Single Condition, and the ones in Dyad Condition were the smallest. In other words, Randomization Condition deviated the most from the expected uniform distribution and the Dyad Condition was the closest.

Griffiths and Tenenbaum extended the idea that people may attempt to generate sequences that are more "representative" of the output of a random process (Kahneman & Tversky, 1974) and built a computational model explaining the deviation from expectation of subjective randomness (Griffiths & Tenenbaum, 2001). I will first reproduce their model to characterize the biases found in Randomization Condition. Then I will extend their model to simulate Dyad Condition and Single Condition. My hypothesis is that people

use Theory-of-Mind reasoning in these two conditions. In Dyad Condition, subjects will use their competitors' previous choices to predict their choice in the current trial, assuming that their competitors intended to generate more "representative" sequences of random process, then they will generate their response accordingly. In Single Condition, subjects would first generate some random choices in their mind, then they would consider that their competitor knew what kind of sequences they are more likely to generate, so they will adjust accordingly. In addition, their own desire to generate random sequences also influenced their choices.

In addition, I also explored an alternative model, the Markov Model, to simulate the three conditions and compare it with the Coin-Weight Model.

Coin-Weight Model

Model Description

Modeling the Randomization Condition. For Randomization Condition, it is the same scenario as described in Griffiths and Tenenbaum's paper. To implement their idea, I combined two Church models, "Randomness Judgements" and "a Communication Game" (Shafto, Goodman, & Frank, 2012). The model assumes that when the subjects were instructed to generate a random sequence, they tried to show the experimenter that the "weight of the coin" was unbiased. This would result in generating sequences that are more representative of a random sequence, such as [0 1 0]. We denote a sequence of length k to be S_k , which can be viewed as a random variable, and a specific instance of it to be s_k ($k = 2, 3, 4$). For instance, S_3 can take values like [0 1 0]. I obtained the probability of $P(S_k = s_k)$ by assuming that the subjects attempted to convince the experimenter that the sequence was randomly generated (Figure 1). The Church model returns the probability of specific k -tuples in Randomization condition such as $P(S_2 = [01])$. A complete collection of the codes implementing models in this paper can be found at <https://web.stanford.edu/~xfyuan/psych204Code>.

Modeling the Dyad Condition For Dyad Condition, the model assumes that subjects would generate responses based on the previous choices made by their competitors and themselves. He would first mentally simulate different alternatives (0 or 1) his competitor might choose in the current trial, and then combine his competitor's previous responses with the current possible responses. He would predict his competitor's current response according to the probability that the combined sequence is judged as random. Mathematically, the probability to choose 0 given the previous responses can be calculated according to equation 1.

```
(define teacher (mem (lambda (seqLength myInten depth)
  (enumeration-query
    (define triedSeq
      (repeat seqLength (lambda ()
        (multinomial '(0 1) '(0.5 0.5))))))
    triedSeq
    (equal? myInten
      (apply multinomial
        (learner seqLength triedSeq depth)))))))

(define learner (mem (lambda (seqLength observedSeq depth)
  (enumeration-query
    (define theWeight
      (uniform-draw
        (list 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9)))
    (define isfair (equal? theWeight 0.5))
    (define coin (lambda ()
      (multinomial '(0 1)
        (list theWeight (- 1 theWeight)))))
    isfair
    (if (= depth 0)
      (equal? observedSeq
        (repeat seqLength coin))
      (equal? observedSeq
        (apply multinomial
          (teacher seqLength isfair (- depth 1)))))))
  (list (second (teacher 2 '#t 1))
        (second (teacher 3 '#t 1))
        (second (teacher 4 '#t 1)))))
```

Figure 1: The Church Codes of the Coin-Weight Model for Randomization Condition.

$$\begin{aligned}
& P(R_k = 0 | S_{k-1} = s_{k-1}) \\
&= \frac{P(R_k = 0 | S_{k-1} = s_{k-1})}{P(R_k = 0 | S_{k-1} = s_{k-1}) + P(R_k = 1 | S_{k-1} = s_{k-1})} \\
&= \frac{P(R_k = 0 \wedge S_{k-1} = s_{k-1})}{P(R_k = 0 \wedge S_{k-1} = s_{k-1}) + P(R_k = 1 \wedge S_{k-1} = s_{k-1})} \quad (1) \\
&= \frac{P(S_k = (s_{k-1}, 0))}{P(S_k = (s_{k-1}, 0)) + P(S_k = (s_{k-1}, 1))}
\end{aligned}$$

With equation 1 and the probability $P(S_k = s_k)$ derived in Randomization Condition we can compute $P(R_k = 0 | S_{k-1} = s_{k-1})$. For instance, assuming that a player's most recent two choices are 0 and 1. The probability that he would choose 0 in the current trial is given by:

$$\begin{aligned}
& P(R_3 = 0 | S_2 = [01]) \\
&= \frac{P(R_3 = 0 | S_2 = [01])}{P(R_3 = 0 | S_2 = [01]) + P(R_3 = 1 | S_2 = [01])} \\
&= \frac{P(R_3 = 0 \wedge S_2 = [01])}{P(R_3 = 0 \wedge S_2 = [01]) + P(R_3 = 1 \wedge S_2 = [01])} \quad (2) \\
&= \frac{P(S_3 = ([01], 0))}{P(S_3 = ([01], 0)) + P(S_3 = ([01], 1))} \\
&= \frac{P(S_3 = [010])}{P(S_3 = [010]) + P(S_3 = [011])}
\end{aligned}$$

$P(S_3 = [010])$ and $P(S_3 = [011])$ are available from the Randomization Condition.

After the subject makes his predictions about his competitor, he will make his choice according to his assigned role, i.e., if he is a matcher, he would choose the same response, if he is a non-matcher, he would choose the

opposite response. In addition, he also cared about whether he himself was generating random sequences. There is a weight term w capturing how subjects integrate these two concerns. The larger the w , the more subjects weigh their competitors' potential choice. In the simulation below, the value of w is set to 0.6. To summarize, the probability to choose 0 given previous responses for a matcher is:

$$\begin{aligned} &P(R_k^M = 0 | S_{k-1}^M = s_{k-1}^M \wedge S_{k-1}^{NM} = s_{k-1}^{NM}) \\ &= w * P(R_k^{NM} = 0 | S_{k-1}^{NM} = s_{k-1}^{NM}) + \\ &(1 - w) * P(R_k^M = 0 | S_{k-1}^M = s_{k-1}^M), \end{aligned} \quad (3)$$

and a non-matcher:

$$\begin{aligned} &P(R_k^{NM} = 0 | S_{k-1}^M = s_{k-1}^M \wedge S_{k-1}^{NM} = s_{k-1}^{NM}) \\ &= w * P(R_k^M = 1 | S_{k-1}^M = s_{k-1}^M) + \\ &(1 - w) * P(R_k^{NM} = 0 | S_{k-1}^{NM} = s_{k-1}^{NM}) \end{aligned} \quad (4)$$

Using a concrete example to illustrate how equation 5 and 6 should be applied, we assume that the most recent two choices made by the matcher is 0 and 1, and those two made by the non-matcher is 1 and 1. The probability to choose 0 as a matcher given his and his competitor's previous responses is:

$$\begin{aligned} &P(R_3^M = 0 | S_2^M = [01] \wedge S_2^{NM} = [11]) \\ &= w * P(R_3^{NM} = 0 | S_2^{NM} = [11]) + \\ &(1 - w) * P(R_3^M = 0 | S_2^M = [01]), \end{aligned} \quad (5)$$

and as a non-matcher:

$$\begin{aligned} &P(R_3^{NM} = 0 | S_2^M = [01] \wedge S_2^{NM} = [11]) \\ &= w * P(R_3^M = 1 | S_2^M = [01]) + \\ &(1 - w) * P(R_3^{NM} = 0 | S_2^{NM} = [11]) \end{aligned} \quad (6)$$

The exact value of equation 5 and 6 can be obtained from equation 1. With those probabilities, I calculated the distribution of all the possible k -tuples and compared the model prediction with the empirical data.

Modeling the Single Condition For Single Condition, the model assumes that subjects would generate the response based on the previous choices made by themselves. For the matchers, they attempted to generate random sequences and they knew their competitors also intended to generate random sequences. To match their competitors' responses, they did not need to correct their biased subjective randomness. However, for non-matchers, they knew their competitors intended to generate random sequences, so their intention to generate random sequences would be suppressed. The weight term for their competitors' perspective is the same w as in Dyad Condition. Mathematically, the probability to choose 0 given previous responses as a matcher is given by equation 7

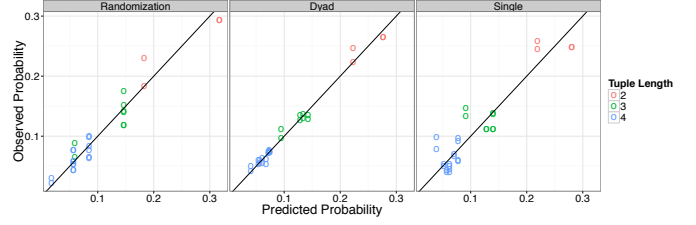


Figure 2: Predictions of the Coin-Weight Model. Left: Randomization Condition. Middle: Dyad Condition. Right: Single Condition.

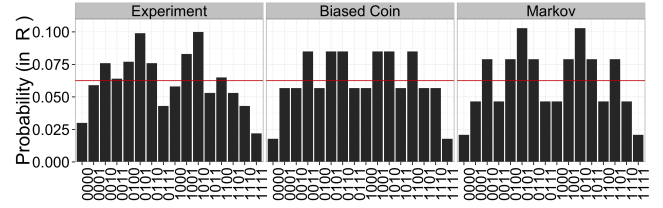


Figure 3: Distributions of 4-tuples in Rapoport and Budescu and model predictions for Randomization Condition. Left: Empirical data. Middle: Predictions of the Coin-Weight Model. Right: Predictions of the Markov Model. The red line indicates expected probability.

$$\begin{aligned} &P(R_k^M = 0 | S_{k-1}^M = s_{k-1}^M) \\ &= w * P(R_k^M = 0 | S_{k-1}^M = s_{k-1}^M) + \\ &(1 - w) * P(R_k^M = 0 | S_{k-1}^M = s_{k-1}^M), \end{aligned} \quad (7)$$

and as a non-matcher:

$$\begin{aligned} &P(R_k^{NM} = 0 | S_{k-1}^{NM} = s_{k-1}^{NM}) \\ &= w * P(R_k^{NM} = 1 | S_{k-1}^{NM} = s_{k-1}^{NM}) + \\ &(1 - w) * P(R_k^{NM} = 0 | S_{k-1}^{NM} = s_{k-1}^{NM}) \end{aligned} \quad (8)$$

Notice that in both equations, the subjects only had access to their own previous responses. The value of weight w was assumed equal to the one in Dyad Condition.

Preliminary Results of the Coin-Weight Model

Randomization Condition Figure 2 Left shows that the model fits the data well, $R^2 = 0.98, p < 0.001$. Figure 3 Middle shows that the model successfully captures the observation that some tuples like (0 0 0 0) are less likely to be generated than some other tuples like (0 1 1 0).

Dyad Condition Although in the Dyad Condition we have different formulae for matchers and non-matchers, the results showed that the distributions of k -tuples are the same. Therefore, I collapsed these two cases (Figure 4 Middle).

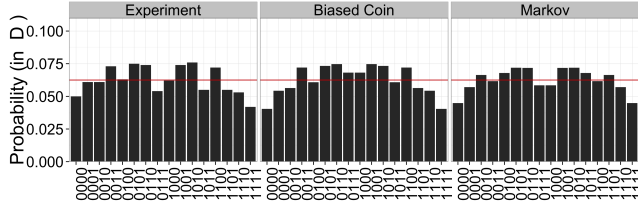


Figure 4: Distributions of 4-tuples in Rapoport and Budescu and model predictions for Dyad Condition. Left: Empirical data. Middle: Predictions of the Coin-Weight Model. Right: Predictions of the Markov Model. The red line indicates expected probability.

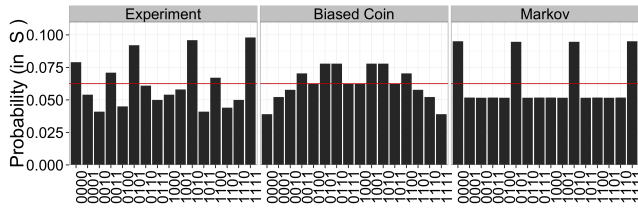


Figure 5: Distributions of 4-tuples in Rapoport and Budescu and model predictions for Single Condition. Left: Empirical data. Middle: Predictions of the Coin-Weight Model. Right: Predictions of the Markov Model. The red line indicates expected probability.

The model predictions are well aligned with the empirical data (Figure 2 Middle), $R^2 = 0.94, p < 0.001$. Critically, it predicts that subjects' biased subjective randomness should be partially corrected as compared with Randomization Condition. The MAD and SD calculated from model predictions in Dyad Condition are much smaller than in Randomization Condition.

Single Condition In the Single Condition, we have different formulae for matchers and non-matchers, and the results showed that the distributions of k -tuples are dramatically different. Rapoport and colleagues did not mention this issue and they just collapsed the two cases. After I averaged the predictions for matchers and non-matchers, I found that the overall performance of the model seems to be fine (Figure 2 Right), but it fails to capture the empirical observation that the probability to generate sequences like (0 0 0 0) or (1 1 1 1) was pretty high.

Markov Model

As seen from the above model fitting results, the Coin-Weight Model does not work very well for the Single Condition. Notice that in the empirical data, both sequences that are more representative of the outcome of tossing an unbiased coin (e.g., [0 1 0 1]) and the less representative ones (e.g., [0 0 0 0])

have higher probabilities. It is very hard for the Coin-Weight Model to reconcile this conflict. Rapoport and Budescu also mentioned in their paper that although the results of Randomization Condition resembled Dyad Condition (the biased subjective randomness was calibrated), it did not imply participants used the same strategy. They further calculated the numbers of alternations in each condition, and found a much higher variance between participants in Single Condition, with a large minority generating sequences with either too many or too few alternations. To capture this finding, I explored an alternative model, the Markov Model, which assumes that each participants generated the binary sequence with an internal transition probability, i.e., $P(R_k \neq R_{k-1})$. I next explain how this model simulates each condition in the current experiment.

Model Description

Modeling the Randomization Condition Similar to the Coin-Weight model, the Markov Model assumes that the subjects tried to convince the experimenter that their sequences were generated by a random process. However, the definition they used for "a random process" is not "tossing an unbiased coin", but a generative process with transition probability of 0.5. If there is some bias in the generating process, the transition probability should be less than 0.5, resulting in more sequences with fewer alternation like [0 0 0 0] and [1 1 1 1]. As in the Coin-Weight Model, we denote the a sequence of length k to be S_k , and a specific instance of it to be s_k ($k = 2, 3, 4$). Using the enumeration-query in Church, I obtained the probability of $P(S_k = s_k)$ given that the subject attempted to convince the experimenter that the transition probability of the underlying generative process equals to 0.5.

Modeling the Dyad Condition For Dyad Condition, the Markov model is very similar to the Coin-Weight Model. It assumes that subjects believed that their competitors intended to generate sequences that simulated the outcome of a generative process with transition probability of 0.5. Therefore, they used their competitors' previous choices to predict their current choice and acted accordingly. Also, they needed to balance this Theory-of-Mind reasoning with their own desire to generate sequences that simulate the outcome of a generative process with transition probability of 0.5. Mathematically, the probability to choose 0 given the previous responses can be calculated using the same formula 1, but now $P(S_k = s_k)$ was obtained by the predictions of Markov Model for the Randomization Condition. We can then use formulae 5 and 6 to calculate the probability to choose 0 as a matcher and a non-matcher given their and their competitors' previous responses. In addition, since the Markov Model and the Coin-Weight have different assumptions, there is no reason that the weight term w assigned to the Theory-of-Mind reasoning should be equal in these two models. Therefore, w was set to 0.2 in the Markov Model to fit the data.

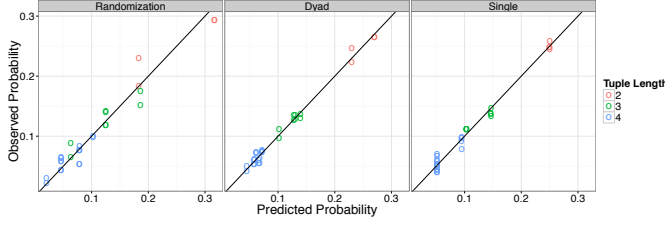


Figure 6: Predictions of the Markov Model. Left: Randomization Condition. Middle: Dyad Condition. Right: Single Condition.

Modeling the Single Condition As mentioned before, subjects' strategy in the Randomization Condition might be very different from the Dyad Condition. To capture the observation that there was a large minority generating sequences with either too many or too few alternations, I assume that there is uncertainty about subjects' transition probabilities and I estimate the posterior distribution of different transition probabilities conditioning on the data. Three levels of transition probability were used, i.e., low(0.25), medium(0.5) and high(0.75). The model first assumes a flat prior on the distributions of these three transition probabilities, then it generates 4-tuples according to these transition probabilities. Conditioning on the empirical distribution of different 4-tuples, the model learned the posterior distribution of these three transition probabilities. Finally, the model makes predictions by sampling the transition probability from its posterior to generate k -tuples. See online codes for more details.

Preliminary Results of Markov Model

Randomization Condition Figure 6 Left shows that the model fits the data well, $R^2 = 0.95, p < 0.001$. Figure 3 Right shows that the model successfully captures the observation that some tuples like [0 0 0 0] are less likely to be generated than some other tuples like [0 1 1 0], but the Markov Model does not show any advantage over the Coin-Weight Model.

Dyad Condition Although in the Dyad Condition we have different formulae for matchers and non-matchers, the results showed that the distributions of k -tuples are the same. Therefore, I collapsed these two cases (Figure 4 Right). The model predictions are well aligned with the empirical data (Figure 6 Middel), $R^2 = 0.99, p < 0.001$. Critically, it predicts that subjects' biased subjective randomness should be partially corrected (see the MAD and SD section).

Single Condition The estimated posterior distribution of the transition probability is $P_{\text{low}} = 0.34, P_{\text{medium}} = 0.31, P_{\text{high}} = 0.35$. The model prediction generated according to this posterior distribution fits the data well (Figure 6 Right), $R^2 = 0.98, p < 0.001$. Critically, it captures the findings that both sequences that are more representative of the outcome

of tossing an unbiased coin (e.g., [0 1 0 1]) and the less representative ones (e.g., [0 0 0 0]) have higher probabilities, because the probability to have a low transition rate and a high transition rate are both higher than the medium transition rate 0.5.

Model Comparison From the correlation plot we see that Markov Model seems to fit the data better than the Coin-Weight Model. However, since these two models have different assumptions, traditional statistical tests for model comparison are not applicable. Therefore, I used a R package called "cocor" to directly compare the correlations between the empirical data and model predictions of these two models (Diedenhofen & Musch, 2014). The results of all tests lead to the convergent conclusion that the difference between the two correlations r_{CW} and r_M is significant, and the null hypothesis should be rejected, $p < .001$. In other words, the Markov Model provides a better fit to the data than the Coin-Weight Model. Therefore, in the following section, I only presented the MAD and SD calculated from the predictions of the Markov Model.

MAD and SD Compared with the data, the model predicts the same qualitative results for MAD and SD of the three conditions (Table 1 and 2), i.e., Randomization Condition has the largest deviation and Dyad Condition has the smallest ones. This suggests that the Markov Model successfully captures the less biased subjective randomness in Dyad Condition.

Table 1: Mean absolute deviation (MAD) of data and the predictions of the Markov Model.

	Data			Model		
	D	S	R	D	S	R
2 – tuple	0.0150	0.0043	0.0435	0.0200	0.0002	0.0667
3 – tuple	0.0105	0.0135	0.0273	0.0115	0.0216	0.0309
4 – tuple	0.0087	0.0159	0.0174	0.0070	0.0162	0.0224

Table 2: Standard deviation (SD) around expectations of data and the predictions of the Markov Model.

	Data			Model		
	D	S	R	D	S	R
2 – tuple	0.0199	0.0057	0.0534	0.0231	0.0003	0.0770
3 – tuple	0.0141	0.0150	0.0353	0.0150	0.0231	0.0465
4 – tuple	0.0105	0.0194	0.0222	0.0088	0.0193	0.0257

Discussion

In the current paper, I explored two probabilistic models to explain the calibrated subjected randomness in Matching

Pennies game as reported in Rapoport and Budescu (1992). I found that although both the Coin-Weight Model and Markov Model are able to capture the calibrated subject randomness in Dyad condition as compared to Randomization Condition, the Markov Model is better than the Coin-Weight Model in explaining the Single Condition. The current modeling study reveals several key properties of human subjective randomness.

First of all, why the Markov Model is better than the Coin-Weight Model in explaining the Single Condition? The reason might be that the transition probability of a generative process is more mentally accessible than "the weight of the coin". When people attempt to generate random sequences, it might be easier to keep track of the transition probability and make sure it approximates 0.5 than to constantly check whether one of the binary responses is made more often than the other. In short, transition probability might be a more convenient heuristic than "the weight of the coin" in evaluating the randomness of sequences.

In addition, it is worth noting that although the Markov Model for Single Condition fits the data relatively well, it is essentially a descriptive model. Therefore, it is fundamentally different from the Coin-Weight Model and the Markov Model for Dyad Condition and Randomization Condition, which are more like mechanistic models. To be specific, the Markov Model for Single Condition estimates the posterior of different transition probabilities by fitting the data, so it does not give any explanation why people have such posterior. By contrast, the Coin-Weight Model and the Markov Model for Randomization Condition assume that subjects generate binary sequences to convince the experimenter that the sequences were generated by a random process, and the resulted distributions of k -tuples naturally exhibit patterns consistent with the data. These two models do not explicitly assume that subjects will generate sequences that are more representative of a random sequence. Instead, this phenomenon spontaneously emerges when subjects try to convince the experimenter that the sequences were randomly generated. Future work is needed to understand the mechanisms underlying the bimodal posterior distribution of different transition probabilities revealed by the Markov Model for Single Condition.

Also, the two models include several parameters that are predetermined. Ideally, they should be estimated using Bayesian data analysis. The critical parameters include the weight w_1 assigned to the Theory-of-Mind reasoning in the Coin-Weight Model, the weight w_2 in the Markov Model for Dyad Condition, and the range of prior transition probability, which is set to be [0.25, 0.75] in the current simulation. The model might be further improved when we estimate the parameters using the data.

Besides, both the Coin-Weight Model and the Markov Model assume that subjects are rational agents and they use the optimal strategy, i.e., generating binary responses as randomly as possible. Therefore, the models cannot

predict a player's behavior when his competitor does not use the optimal strategy. For example, if matcher "A" plays with a person who chooses "0" far more often than "1", A would quickly notice it and choosing "0" more (or always). However, the two models in the current study is not able to predict that because the assumption that the other agent intends to generate random sequences is "hardwired". In this case, more complicated algorithm such as reinforcement learning might be more suitable to simulate the task (Lee, Conroy, McGreevy, & Barraclough, 2004), and it is worth comparing the assumptions and predictions of the current probabilistic approach with previous reinforcement learning approach.

In summary, the current study provides a new computational perspective for interpreting the behavioral data of subjective randomness in competitive games and sheds light on the mechanisms underlying the generation of random sequences when feedback is available.

References

- Bakan, P. (1960). Response-tendencies in attempts to generate random binary series. *The American journal of psychology*, 127–131.
- Diedenhofen, B., & Musch, J. (2014). cocor: A comprehensive solution for the statistical comparison of correlations. *PloS one*, 10(3), e0121945–e0121945.
- Griffiths, T. L., & Tenenbaum, J. B. (2001). Randomness and coincidences: Reconciling intuition and probability theory. In *Proceedings of the 23rd annual conference of the cognitive science society* (pp. 370–375).
- Griffiths, T. L., & Tenenbaum, J. B. (2003). From algorithmic to subjective randomness. In *Advances in neural information processing systems* (p. None).
- Kahneman, D., & Tversky, A. (1974). Subjective probability: A judgment of representativeness. In *The concept of probability in psychological experiments* (pp. 25–48). Springer.
- Lee, D., Conroy, M. L., McGreevy, B. P., & Barraclough, D. J. (2004). Reinforcement learning and decision making in monkeys during a competitive game. *Cognitive Brain Research*, 22(1), 45–58.
- Neuringer, A. (1986). Can people behave "randomly"? The role of feedback. *Journal of Experimental Psychology: General*, 115(1), 62.
- Rapoport, A., & Budescu, D. V. (1992). Generation of random series in two-person strictly competitive games. *Journal of Experimental Psychology: General*, 121(3), 352.
- Shafra, P., Goodman, N. D., & Frank, M. C. (2012). Learning from others the consequences of psychological reasoning for human learning. *Perspectives on Psychological Science*, 7(4), 341–351.