Does GPT-4 Play Dice?

Qiang Liu[*]
Tencent Inc.

Abstract

OpenAI's Generative Pre-trained Transformer 4 (GPT-4) is a powerful large language model with a certain degree of intelligence in understanding and generating coherent text. We are exploring whether GPT-4 is capable of acting as a die, i.e. generating random numbers. We show that GPT-4 does not appear to generate independent and identically distributed random numbers. Examples imply that GPT-4 tries to compensate for the uniformity of random numbers by sacrificing independence when acting as a die.

1 Introduction

Persaud [Per05] indicates that humans can consciously generate random number sequences. Thus, machine intelligence can be potentially tested by examining the ability to generate random numbers. Late, Figurska et al. [FSK08] replicate Persaud's experiment and lead to the opposite conclusion. Generating random numbers may not be a necessary capability for an intelligent agent, and it might even be contradictory to intelligence since mechanical computer programs generate better random numbers than humans. It is interesting to check whether GPT-4 [Ope23], which shows a high degree of intelligence in understanding and generating human-like text, is capable of generating random numbers or not. Previously, Hopkins et al. [HRC23] show that Large Language Models (LLM) overall underperform in generating random numbers. We find that GPT-4 is no better at generating random numbers than open-source LLMs but behaves differently.

2 Experiment: Generating only one number each time

We first use the following prompt to instruct GPT-4 to mimic playing dice and output a random number:

Prompt 1 *You are playing a game of dice. Please generate a random number between 1 and 6, which represents the outcome of rolling a standard six-sided die.*

We set the top-p and temperature parameters to be 0.95 and 1, respectively. Top-p and temperature should not alter the randomness of the output. We request GPT-4 for 2000 times and obtain 2000 numbers. In each request, GPT returns exactly one integer number between 1 and 6. Figure 1a shows that the result does not follow the uniform distribution and there is no number 1 and number 6 in the outputs. Although we tell GPT to roll a six-sided die, GPT may misunderstand our prompt that *between 1 and 6* equals to set {2, 3, 4, 5}. To make sure GPT receives a clear message, we do the experiment again using the following prompt:

Prompt 2 *You are playing a game of dice. Please generate a random number from [1, 2, 3, 4, 5, 6], which represents the outcome of rolling a standard six-sided die.*

Figure 1b indicates an improvement since GPT outputs number 6. However, the distribution is far from a uniform one. The frequency of each random number seems to be related to the position of the number in prompt 2: The number in the middle appears more frequently. We disorder the numbers in prompt 2 and obtain:

---

[*]Email: Q.L.liu@hotmail.com
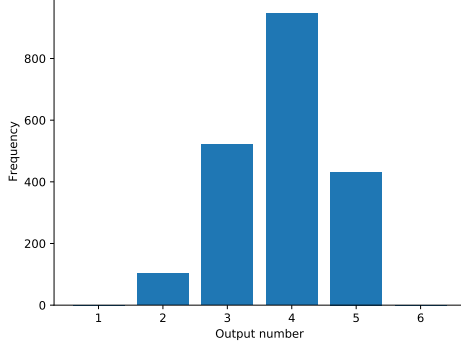
(a) Prompt 1.

(b) Prompt 2.
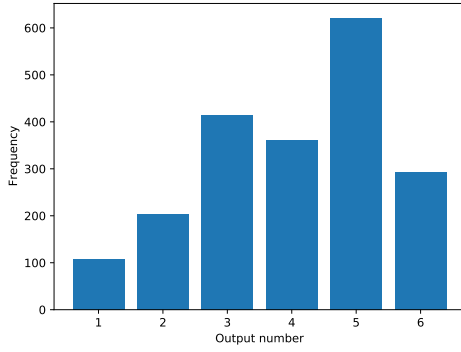
(c) Prompt 3.
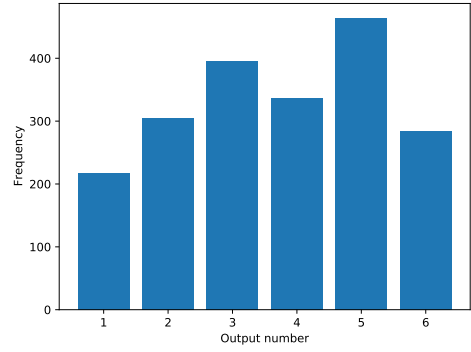
(d) Modified prompt 3.

Figure 1: Frequency of random number generated by GPT-4 using prompt 1 to 4.
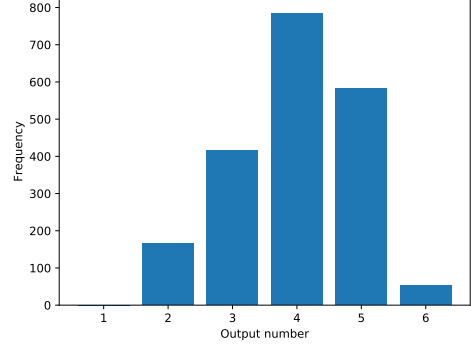
Prompt 3 *You are playing a game of dice. Please generate a random number from [4, 2, 6, 1, 5, 3], which represents the outcome of rolling a standard six-sided die.*

Figure 1c shows the result of using prompt 3. The frequency of generating number 4 (appears first in prompt 3) decreases compared to the results from prompt 1 and 2, but the frequencies of numbers 1 and 6 (in the middle) are still among the lowest which may result from inherent mechanisms of GPT model or training corpus. For GPT-4, the above prompt should be already clear for this simple task, but to be more explicit, we add an extra explanation at the end of prompt 3: *The probability of selecting each number should be equal.* From Figure 1d, the output of modified prompt 3 has a similar pattern compared to that of prompt 3, but is more uniform.

The above-mentioned experiments indicate that the returned number from GPT-4 is not uniformly random.

3 Experiment: Generating random sequence

In section 2, GPT-4 generates only one sample number for each request. We explore whether GPT-4 is capable of generating a sequence of random numbers. We modify prompt 3 and obtain the following prompt:

Prompt 4 *You are playing a game of dice. Please generate a random number sequence with 280 numbers. Each number is selected independently from [4, 2, 6, 1, 5, 3], which represents the outcome of rolling a standard six-sided die. The probability of selecting each number should be equal.*

Prompt 4 does not always instruct GPT-4 to generate sequence but instead output *Python* code which generates random numbers if the sequence length is largely set. This can be prevented by explicitly declaring: *Do not generate program code* or reducing the sequence length in the prompt.

The length of the generated sequence does not always follow our instructions. Three generated sequences are as follows:

Sequence 1: *2, 6, 2, 4, 5, 2, 1, 3, 4, 1, 2, 4, 5, 6, 2, 3, 5, 6, 3, 5, 1, 2, 4, 6, 3, 2, 5, 4, 6, 2, 1, 5, 6, 3, 1, 6, 5, 4, 2, 3, 4, 1, 2, 5, 3, 2, 4, 6, 3, 2, 1, 5, 3, 6, 4, 2, 3, 4, 6, 1, 6, 4, 3, 2, 6, 5, 2, 3, 5, 4, 3, 2, 4, 5, 6, 1, 5, 6, 3, 4, 1, 3, 5, 6, 2, 1, 5, 3, 6, 2, 4, 5, 2, 6, 4, 1, 3, 2,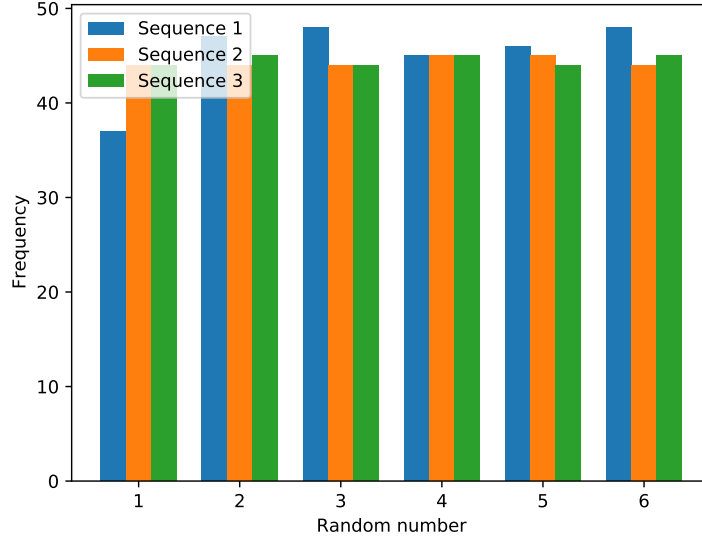 5, 4, 3, 6, 5, 4, 1, 3, 6, 2, 5, 4, 1, 6, 3, 5, 2, 6, 3, 1, 2, 4, 3, 6, 1, 5, 3, 4, 6, 2, 5, 1, 4, 3, 5, 2, 6, 4, 1, 3, 5, 6, 4, 2, 6, 3, 5, 4, 1, 2, 6, 5, 4, 3, 6, 2, 1, 4, 6, 5, 3, 1, 5, 4, 2, 6, 3, 5, 4, 2, 1, 3, 4, 6, 5, 2, 1, 6, 3, 5, 2, 4, 1, 3, 4, 2, 6, 5, 1, 3, 2, 4, 6, 5, 1, 2, 6, 4, 3, 5, 6, 4, 1, 2, 3, 4, 5, 1, 6, 3, 2, 5, 6, 4, 3, 1, 5, 2, 3, 4, 6, 2, 5, 6, 1, 3, 2, 4, 5, 3, 2, 6, 1, 5, 3, 6, 4, 1, 2, 4, 5, 3, 6, 1, 3, 4, 6, 2, 5, 1, 3, 4, 5, 2, 6, 3, 1, 2, 4, 5, 3, 6, 1, 2, 6, 3, 5, 4, 6, 1, 2, 5, 3.*

Sequence 2: *157, 2, 5, 4, 6, 1, 1, 3, 2, 5, 6, 4, 4, 5, 2, 3, 1, 6, 6, 3, 5, 2, 1, 4, 2, 1, 4, 3, 6, 5, 5, 2, 4, 3, 6, 1, 3, 5, 6, 4, 2, 1, 3, 6, 5, 1, 4, 2, 6, 1, 4, 3, 2, 5, 3, 1, 5, 4, 6, 2, 2, 6, 1, 4, 5, 3, 6, 1, 5, 2, 3, 4, 1, 5, 3, 6, 2, 4, 5, 6, 4, 1, 3, 2, 4, 3, 6, 1, 2, 5, 6, 3, 4, 5, 2, 1, 3, 6, 1, 4, 2, 5, 4, 6, 5, 1, 2, 3, 2, 5, 6, 4, 1, 3, 6, 4, 1, 3, 2, 5, 3, 5, 1, 6, 4, 2, 1, 5, 3, 6, 2, 4, 4, 6, 5, 1, 3, 2, 4, 3, 5, 6, 1, 2, 6, 1, 3, 5, 4, 2, 4, 3, 5, 2, 1, 6, 5, 6, 4, 2, 1, 3, 1, 4, 5, 2, 3, 6, 4, 5, 1, 6, 2, 3, 3, 2, 1, 4, 6, 5, 1, 4, 3, 2, 5, 6, 1, 2, 5, 4, 3, 6, 5, 2, 1, 4, 3, 6, 1, 6, 4, 3, 5 , 2, 2, 1, 5, 3, 4, 6, 6, 5, 3, 1, 4, 2, 4, 1, 6, 5, 2, 3, 4, 6, 3, 2, 1, 5, 5, 6, 1, 3, 4, 2, 6, 2, 4, 5, 3, 1, 1, 2, 6, 4, 3, 5, 2, 3, 1, 5, 6, 4, 5, 1, 2, 3, 4, 6, 3, 6, 5, 1, 4, 2, 5, 4, 3.*

Sequence 3: *6, 1, 4, 2, 5, 3, 4, 6, 2, 1, 3, 5, 3, 6, 1, 4, 2, 5, 4, 3, 6, 1, 2, 5, 3, 2, 4, 6, 1, 5, 2, 4, 3, 6, 5, 1, 3, 4, 2, 6, 5, 1, 4, 3, 6, 2, 5, 1, 4, 6, 3, 2, 1, 5, 4, 3, 6, 2, 5, 3, 1, 6, 4, 2, 1, 5, 6, 3, 4, 2, 1, 5, 3, 6, 4, 2, 1, 6, 5, 3, 4, 2, 6, 1, 5, 3, 4, 2, 1, 5, 6, 4, 3, 2, 5, 1, 6, 4, 2, 3, 1, 5, 6, 4, 2, 3, 6, 1, 5, 4, 2, 3, 1, 6, 5, 4, 2, 3, 6, 1, 4, 5, 2, 3, 1, 4, 6, 5, 2, 1, 3, 4, 6, 5, 2, 3, 1, 5, 6, 4, 2, 3, 1, 4, 5, 6, 2, 1, 3, 4, 6, 5, 2, 3, 1, 4, 5, 6, 2, 1, 4, 3, 5, 6, 1, 2, 3, 4, 5, 6, 2, 1, 4, 3, 5, 6, 1, 2, 4, 3, 5, 6, 1, 2, 3, 4, 6, 5, 1, 2, 4, 3, 5, 6, 1, 2, 4, 3, 5, 6, 2.*

In sequence 2, there is an exception number 157 which does not follow our prompt and we omit this error. Figure 2 shows that the distribution is more uniform compared to the independently generated numbers in section 2. However, all the sequences are not independent. We evaluate the correlation among the random numbers in each sequence $[x_1, x_2, \cdots]$ by counting the identical number pairs $|\{x_i = x_j | d = i - j\}|$ for each distance $d = i - j$, where $i > j$. For an ideal random sequence with length N, the expectation of the number of pairs is

$$E[|\{x_i = x_j | d = i - j\}|] = \frac{N - d}{6}. \tag{1}$$

Equation 1 holds because the number of pairs with distance d is $N - d$ and the probability of the two numbers being identical is $1/6$. As shown in figure 3, the black curve shows the ideal number of identical pairs with each distance d by Eq. 1, and the blue curve represents a real random number sequence that oscillates around its expectation. For sequence 1 and sequence 2, the curves are similar to that of the random one, but at distance $d = 1$ the number of identical pairs deviates largely from the expectation. Especially for sequence 1, $|\{x_i = x_j | i - j = 1\}| = 0$ indicates that all the adjacent numbers are not equal. GPT-4 seems to avoid generating an identical random number to the previous one to maintain the uniformity of random numbers. For sequence 3, the distance correlation shows a period oscillation and GPT-4 is repeating sub-sequences to build the long sequence with the constraint of uniformity of random numbers in our prompt.

4   Conclusion

Our primary experiment shows that GPT-4 cannot generate independent and identically distributed random numbers as a die and compensates for the uniformity of randomness by sacrificing independence. However, generating random numbers may not be a necessity for an intelligent agent, such as a human [FSK08], in tasks like decision-making and problem-solving. Nevertheless, humans can generate high-quality pseudo-random numbers by applying algorithms. If a large model is intelligent enough, the model should be capable of applying these algorithms, too.

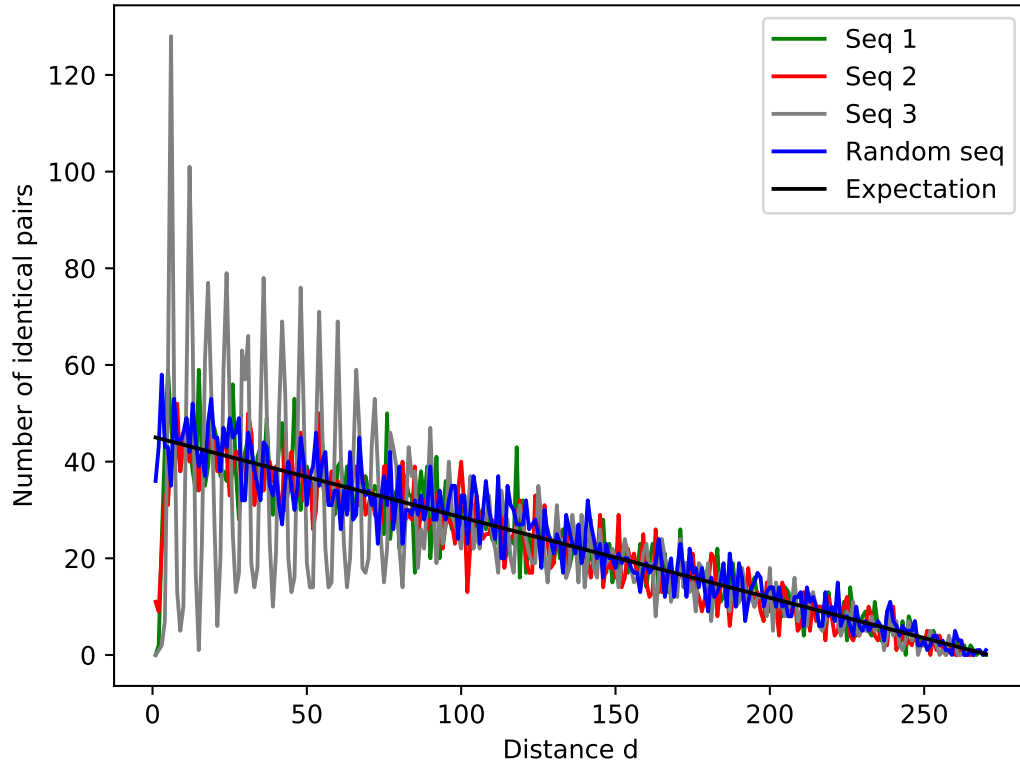Figure 2: Frequency of random number generated by GPT using prompt 4.

Figure 3: Correlation of random number generated by GPT using prompt 4.

References

[FSK08] Małgorzata Figurska, Maciej Stańczyk, and Kamil Kulesza. Humans cannot consciously generate random numbers sequences: Polemic study. *Medical hypotheses*, 70(1):182–185, 2008.

[HRC23] Aspen K Hopkins, Alex Renda, and Michael Carbin. Can llms generate random numbers? evaluating llm sampling in controlled domains. In *ICML 2023 Workshop: Sampling and Optimization in Discrete Space*, 2023.

[Ope23] OpenAI. Gpt-4 technical report, 2023.

[Per05] Navindra Persaud. Humans can consciously generate random number sequences: A possible test for artificial intelligence. *Medical hypotheses*, 65(2):211–214, 2005.