Increasing Don 2 Randomness

lucandor158

August 12, 2023

1 Introduction

Throughout the lifetime of the MTPO community, we have seen that many random events do not behave randomly when playing through the game as normal. There have been many times players have tracked statistics and have received the worst 1/1000 luck or similar. This shows that people can "manipulate" the RNG without knowing resulting in worse luck. This is a somewhat well understood phenomenon with our current understanding of RNG. Many random events are not independent, so when going for strategies that require many aligning random events, the rates are not what they "should" be.

A distinct phenomenon which has been popular among players is that certain players get better or worse luck on certain fights based on the inputs they do. This is also obviously true in certain cases: i.e. someone who knows how to do hippo manipo vs. someone who plays hippo normally. These 2 players are sampling from different random distributions when they get a time on hippo. The hypothesis is that this also happens without any intentional manipulation, just by the nature of how different players do the fight.

The proposed mechanism is that players draw their inputs for a fight from separate random distributions. Even when doing the same strategy, some players will have a tendency to hold the button on a certain punch for longer, or delay a longer time before the next punch. Since the RNG depends on the inputs, this results in separate distributions for different players.

Below we investigate the idea that a certain player has the tendency to have their inputs to a fight be extremely consistent. We investigate what these theoretical extremely consistent players can do to return to randomness.

2 Increase Don 2 Randomness

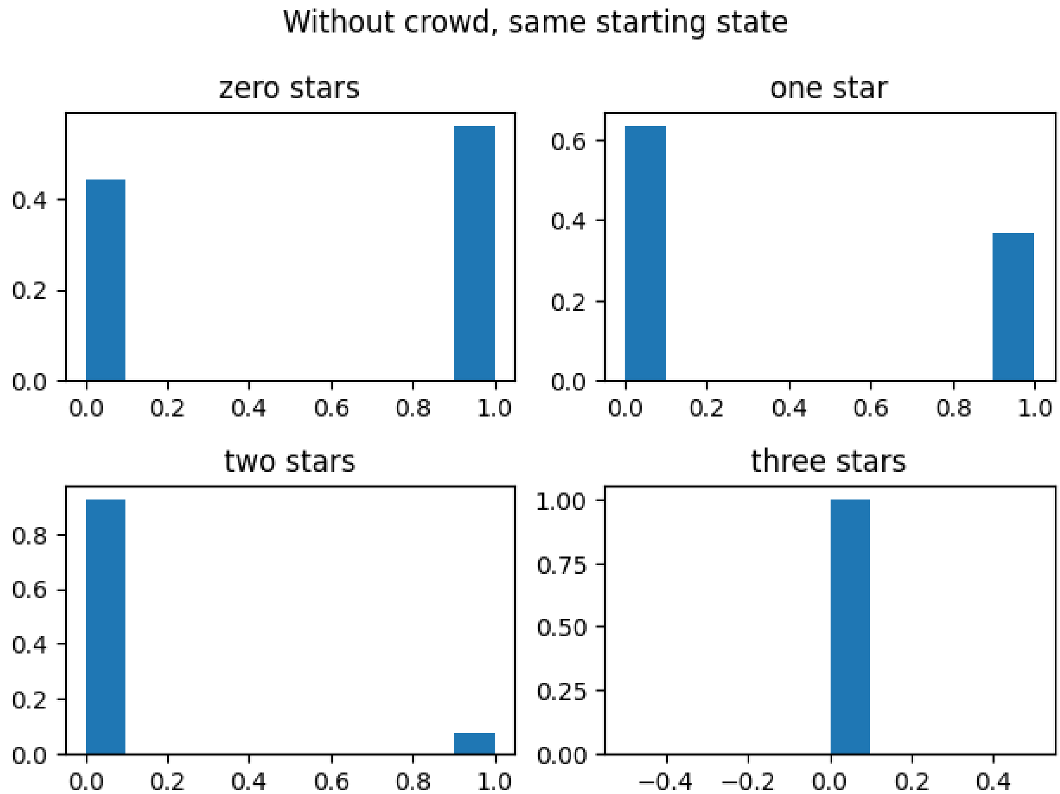
We have a model that can simulate a playthrough of don 2 phase 1. Give some parameters, $p \in P$, and it will tell you how many stars you got. This is a sample from the modeled random distribution. Repeat n times and you can calculate the sample proportion of getting 0, 1, 2, or 3 stars. Let $\hat{p}_{i,p}$ be the sample proportion of getting i stars using parameters $p \in P$.

Let's define parameter space P to include the 11 delays between punches, how long you hold A on guts, how long you hold up and B on faces, and how long you hold start on star throws. Some theoretical consistent player plays according to some parameters p that exist in this space P. The player plays the fight the exact same way every time.

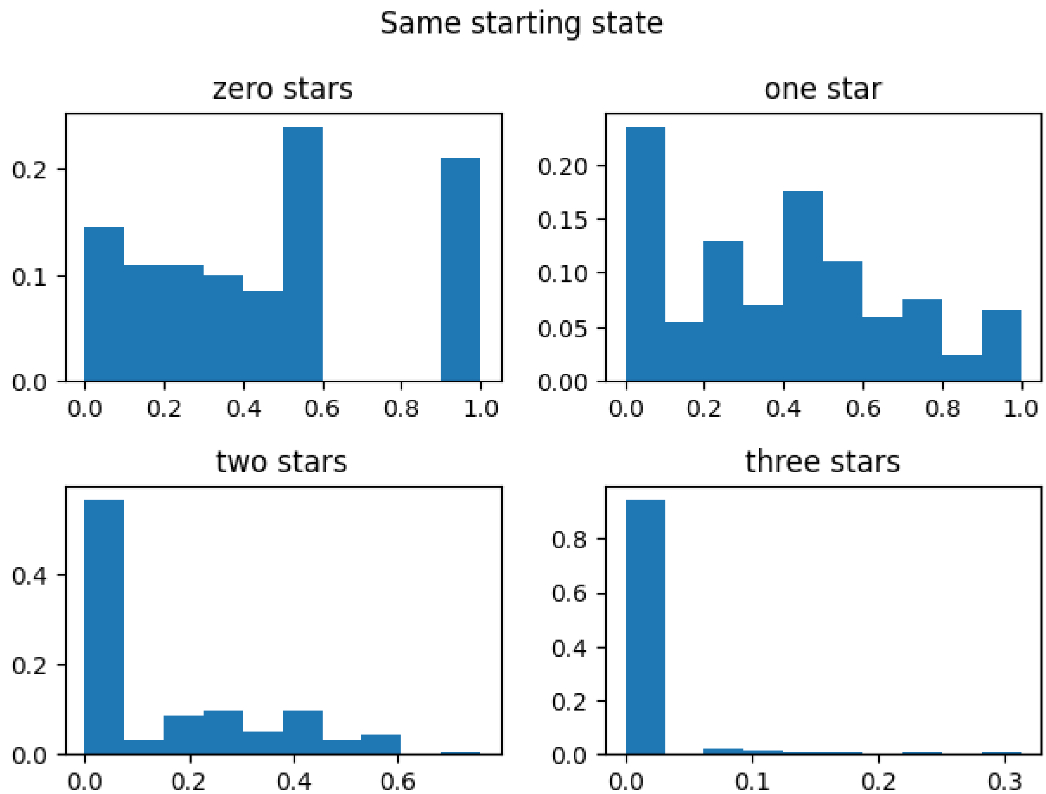
Now the questions we are trying to answer is how much does $\hat{p}_{i,p}$ depend on p, and what techniques can we use to reduce $\hat{p}_{i,p}$'s dependence on p.

3 Simulations

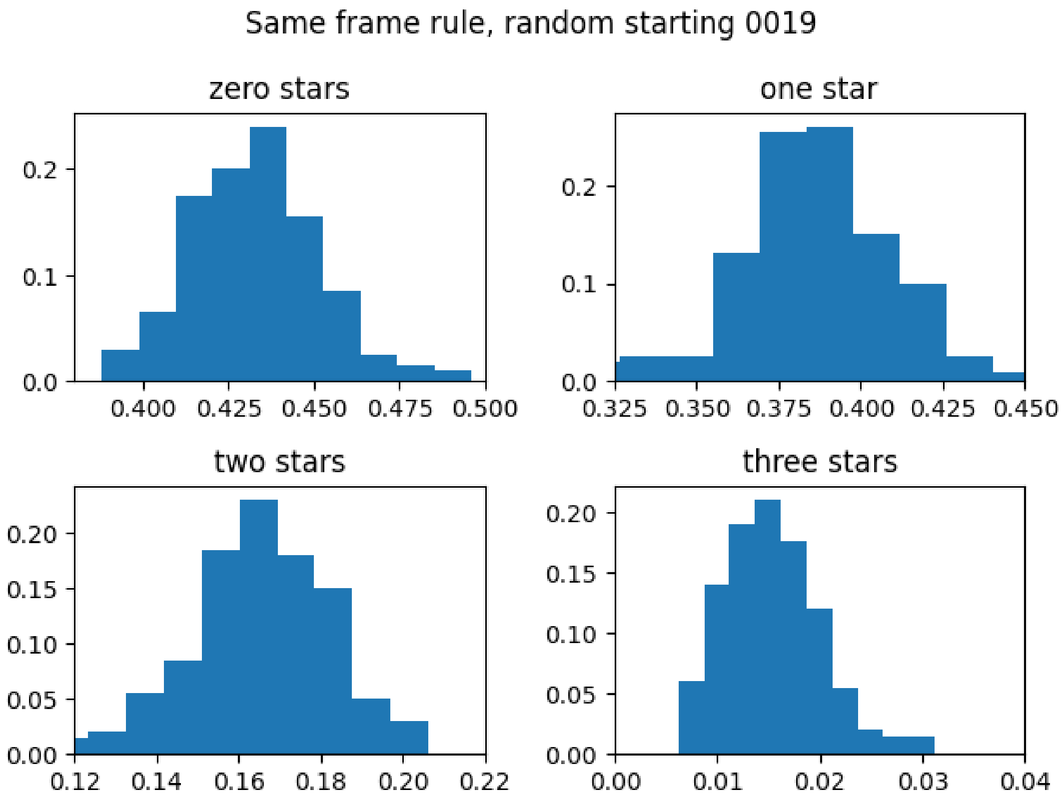
First let's look if we take out the crowd and arrive at a set frame rule (0x001E) and set cumulative inputs (0x0019). Our model now has no randomness and every time for the same parameters we get the same number of stars. This means $\hat{p}_{i,p}$ is either 0 or 1 for all $i \in \{0, 1, 2, 3\}$, $p \in P$.



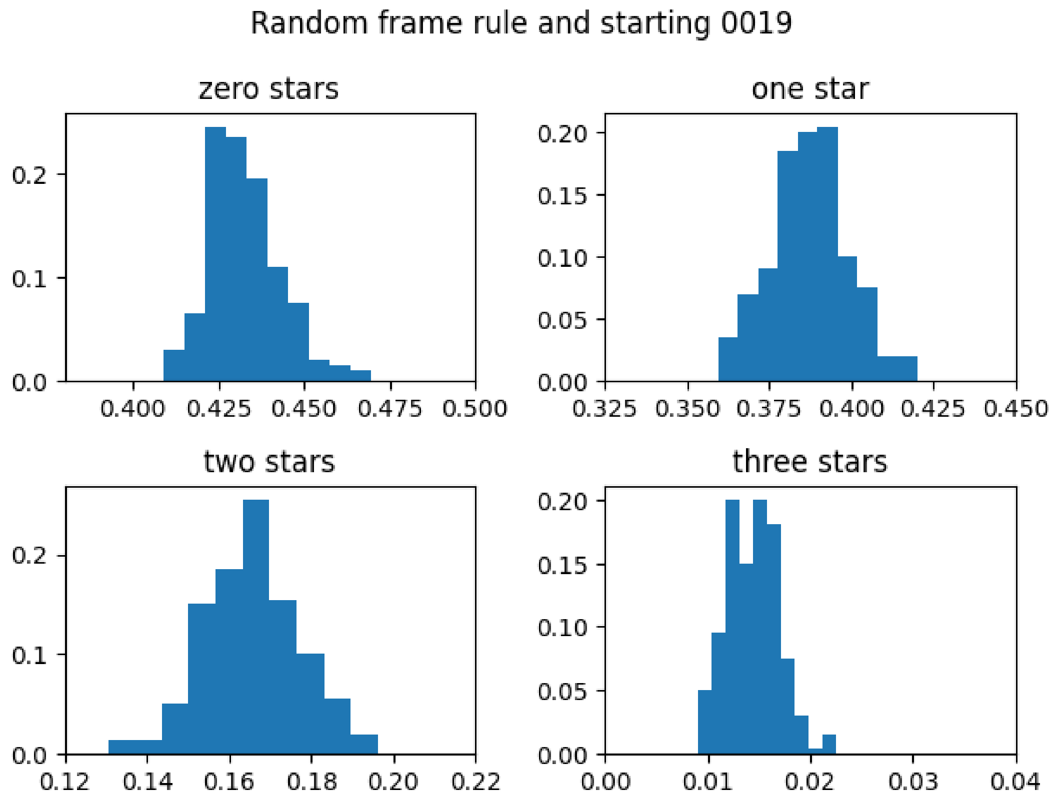
Now let's add the crowd back in. Now we have some randomness outside the player's control.



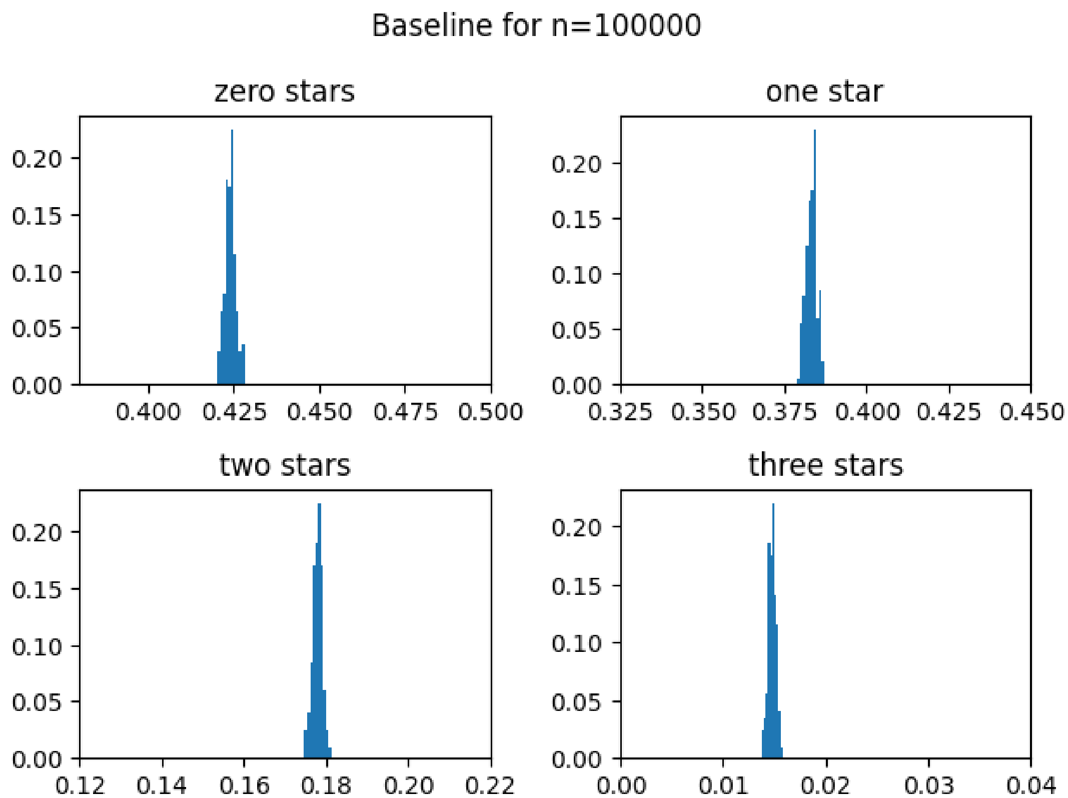
Now let's make the starting input state random (starting 0x0019). This models how we do not know what inputs we have done up to the don 2 fight.



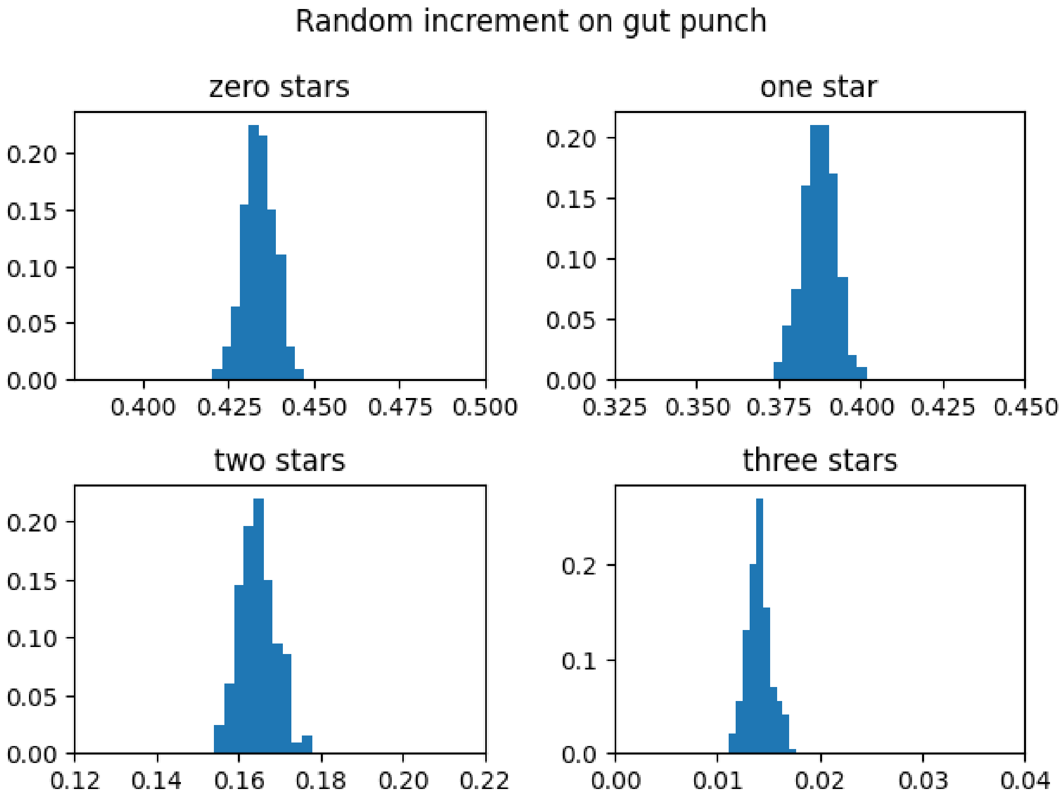
Now let's make the starting frame rule random. This graph is our best approximation for luck of different players with perfectly consistent inputs.



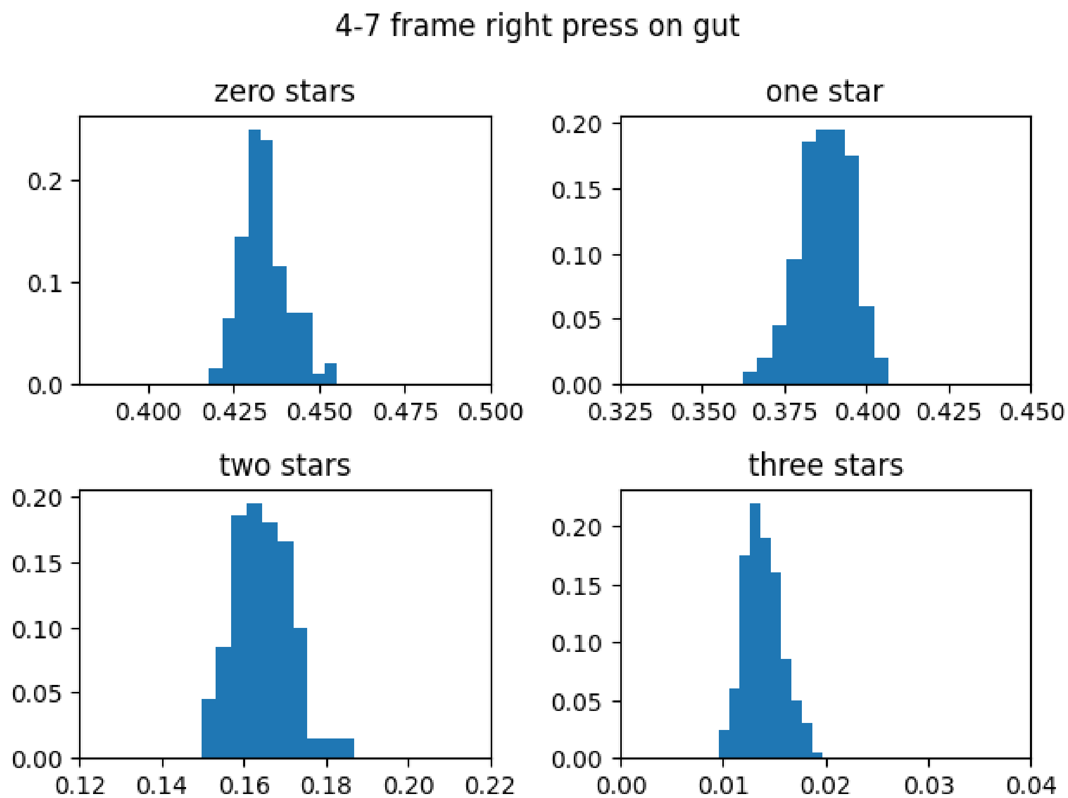
Let's consider how much of this randomness is from not running the simulation enough times (too small n). This is our "baseline" and we cannot get a tighter distribution than this.



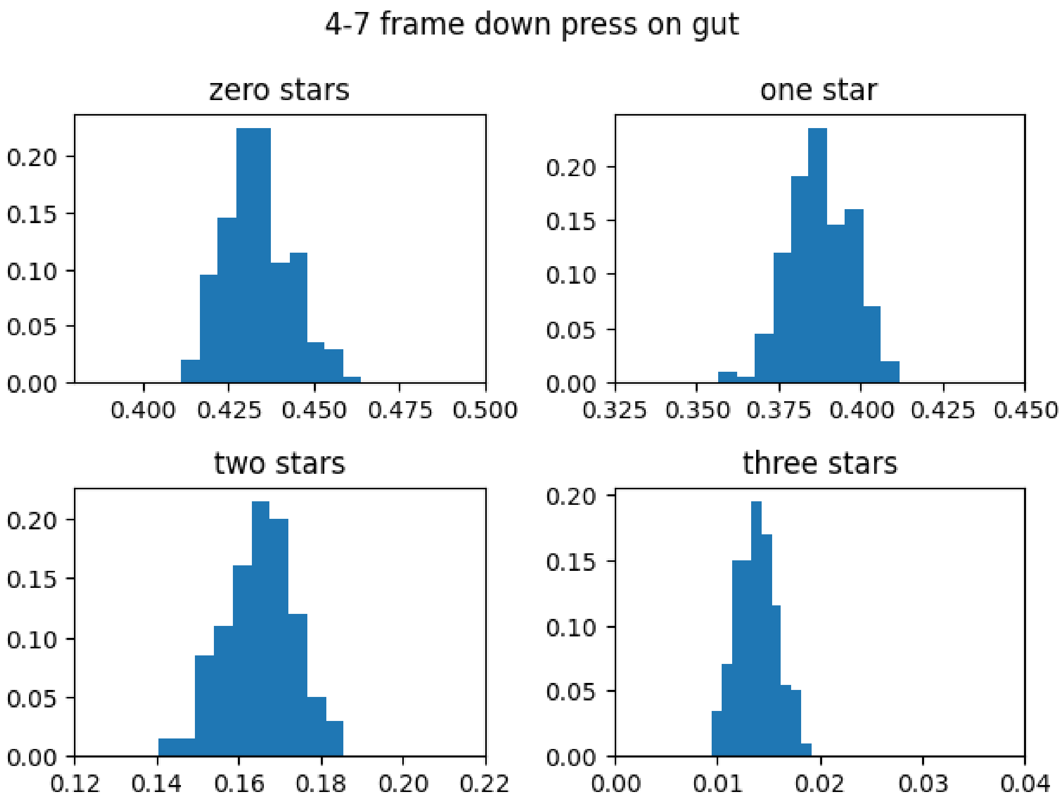
Now let's consider some methods to increase randomness and tighten the distribution for these different players. Imagine we could randomize 0x0019 every time we do a gut punch. That gives the following graph.



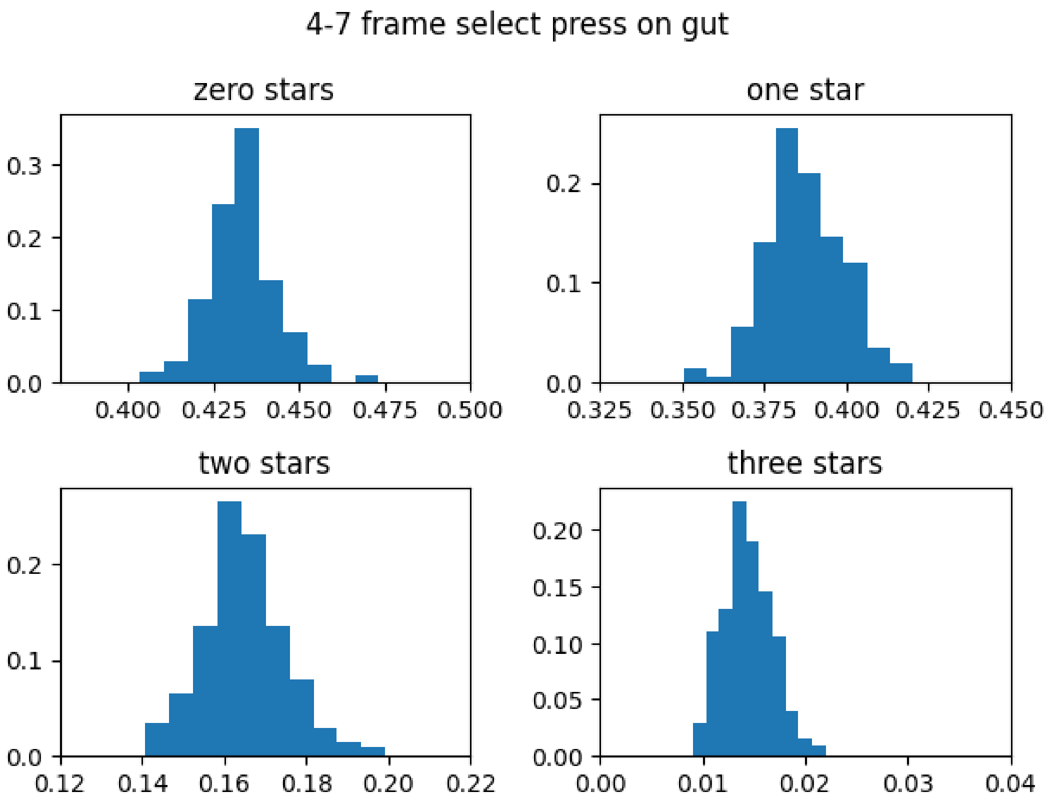
Let's try pressing a certain button for a random number of frames between 4 and 7 on each gut punch. We will do buttons that can be easily pressed while doing a punch. First let's try right.



Next down.

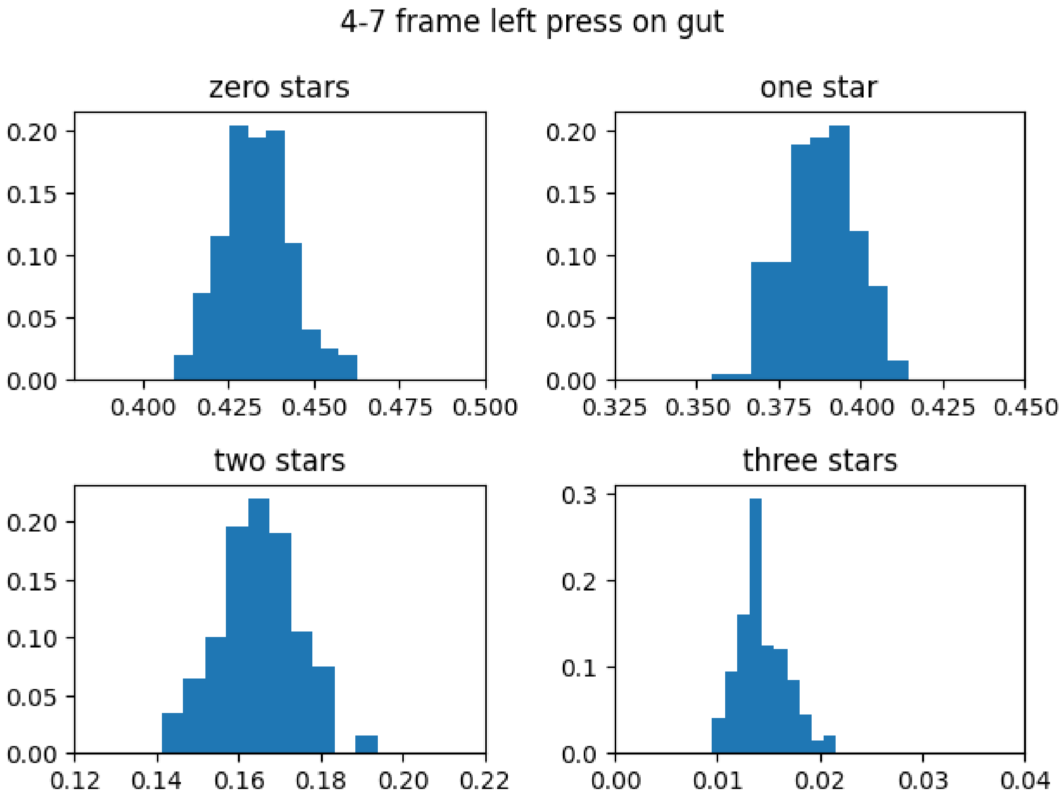

4-7 frame down press on gut
zero stars
one star
two stars
three stars
0.00
0.05
0.10
0.15
0.20
0.400
0.425
0.450
0.475
0.500
0.325
0.350
0.375
0.400
0.425
0.450
0.12
0.14
0.16
0.18
0.20
0.22
0.00
0.01
0.02
0.03
0.04

Next select.



4-7 frame select press on gut
zero stars
0.3
0.2
0.1
0.0
0.400
0.425
0.450
0.475
0.500
one star
0.2
0.1
0.0
0.325
0.350
0.375
0.400
0.425
0.450
two stars
0.2
0.1
0.0
0.12
0.14
0.16
0.18
0.20
0.22
three stars
0.20
0.15
0.10
0.05
0.00
0.00
0.01
0.02
0.03
0.04

Next left.


4-7 frame left press on gut
zero stars
one star
two stars
three stars
0.20
0.15
0.10
0.05
0.00
0.400
0.425
0.450
0.475
0.500
0.325
0.350
0.375
0.400
0.425
0.450
0.12
0.14
0.16
0.18
0.20
0.22
0.3
0.2
0.1
0.0
0.00
0.01
0.02
0.03
0.04

Let's visualize the standard deviations in a chart. This shows how much different strategies increase the randomness.

	0 star st dev	1 star st dev	2 star st dev	3 star st dev
Without crowd	0.4963	0.4814	0.2633	N/A
Same starting state	0.3356	0.2985	0.1884	0.0355
Same frame rule	0.0187	0.0225	0.0169	0.0047
Random starting state	0.0104	0.0118	0.0115	0.0025
Baseline	0.0015	0.0016	0.0011	0.0003
Random inc on gut	0.0046	0.005	0.0044	0.0011
Right on gut	0.0067	0.008	0.0069	0.0018
Down on gut	0.009	0.0102	0.0089	0.0022
Select on gut	0.0098	0.0119	0.01	0.0023
Left on gut	0.0103	0.0126	0.0106	0.0023

4 Conclusion

In modeling theoretical perfectly consistent players, we establish a bound on how much luck depends on your inputs. In the random starting state graph, we see that a player with "ideal" inputs would get 3 stars twice as often as a player with "unideal" inputs. This shows that the way you do inputs on don 2 can matter. However, we see that the spread for getting 0 stars is about 41-47% depending on inputs. This does not explain huge differences in players getting 0 stars much more than others.

It looks like pressing right on a gut punch significantly decreases the variation in different consistent players' luck. It performs better than down, select, or left. This is the recommended approach for now for players looking to increase randomness on don 2.