**Some clever TBD title here**

And also a clever subtitle, if appropriate

**Abstract**

We did lots of great things.

**Introduction**

No Limit Texas Hold’em (NLHE) is a game that utilizes an ordinary 52-card deck of cards. Like most card games, an essential facet to the game of NLHE is that the cards are randomized prior to each round of play, or hand, via shuffling. It has been previously demonstrated that the specific protocol that is generally used by casino poker rooms to shuffle their decks of cards does not sufficiently randomize the cards, to an extent that can be exploited by players to gain an advantage (Merz & Chi, 2022).

The components of the protocol utilized by casino poker rooms are:

1. Riffle shuffle: The deck is split into two portions of approximately equal size, and then the cards are interleaved back together.
2. Cut: The deck is split into two portions, and the portion originally on the bottom gets placed on top of the portion that was originally on top.
3. Strip shuffle: Also known as a box, this is similar to the aforementioned cut but instead of splitting into two portions, the deck is split into some greater number of portions (typically four), and the portions are placed back together in reverse order.

The protocol in question, which we will refer to as the CS (for Casino Shuffle) then consists of the following, in this order: two riffle shuffles, one strip shuffle, one riffle shuffle, and one cut.

In this work, we investigate an additional shuffling technique that is sometimes utilized prior to the CS, known as the wash. To perform a wash, the dealer spreads out all 52 cards in an arc in front of them, and then rotates each hand in opposing circular motions. As the hands move, cards move with them, thus changing the cards’ relative positions until finally the cards are gathered back together. The primary question we aim to answer in this manuscript is: when a wash is performed prior to the CS, how much does this impact the exploitability of the CS by Texas Hold’em players as demonstrated by Merz & Chi (2022)?

Enroute to an answer to this question, we first develop a probabilistic model of the wash shuffle. Unlike many other shuffling techniques, the wash shuffle has not yet been well-studied or modeled. Our development of such a model is thus an additional contribution to the literature. We then utilize this model to simulate wash shuffles prior to the CS, and investigate the resulting impact on the distribution of cards as they are dealt for Texas Hold’em. We find that xxxxxx

**Literature Review**

The primary component of the CS is the riffle shuffle, which has been studied extensively in the literature. A probabilistic model for the riffle shuffle was first proposed by Gilbert (1955) and then independently by Reeds (1981), and has since been dubbed the Gilbert-Shannon-Reeds (GSR) model. Notably, it has been demonstrated that the GSR model produces deck orderings that accurately reflect what occurs when riffle shuffling is performed by a human (Diaconis, 1988). The GSR model prescribes the following to be performed:

1. The deck is split into two portions, with the dividing point determined according to a Binomial(n=52, p=0.5) distribution. Let A be the size of the top portion and B be the size of the bottom portion.
2. The portions are separated, and cards are dropped probabilistically from each portion into position to form the newly ordered deck. The probability that the first card to drop is from each portion is A/(A+B) and B/(A+B) for portions A and B respectively. After the first card drops, A or B changes its value accordingly, and thus the probabilities update for the next card. Cards continue to drop and the probabilities continue to update, until all cards from both portions have dropped into place.

Utilizing the GSR model, Aldous and Diaconis (1986) investigated how quickly, in terms of the number of iterations of a riffle shuffle, an unshuffled deck moves in distribution towards that of a perfectly shuffled deck according to the total variation distance metric. A perfectly shuffled deck is defined as one in which every possible ordering of the 52 cards has a 1/52! probability. Their investigation corroborated earlier work that recommends seven riffle shuffles to sufficiently randomize 52 cards (Aldous, 1983).

However, the CS only contains a total of three riffle shuffles, in addition to a strip shuffle and a cut. To the best of our knowledge, the CS is universally used in casino poker rooms and in major poker tournaments across the world; some examples of professional dealers using the CS are in the 2017 and 2022 World Series of Poker (WSOP), in footage that is available freely online (PokerGO, 2017; PokerGO, 2022). Further evidence as to the ubiquity of the CS was mentioned in (Merz & Chi, 2022), in which it was also demonstrated that the CS leaves a deck in configurations that are exploitable by Texas Hold’em players.

One aspect that was left uninvestigated by Merz & Chi (2022) is the wash shuffle, in which the cards are spread out in front of the dealer and scrambled at random (see e.g. Black, 2020 for a video demonstration). A wash shuffle does not always occur prior to each hand of play; typically, a wash will be performed when a new box of cards is opened, or in some other circumstances such as when a dealer change occurs, or when an error has occurred during the CS process, e.g. as indicated in a 2013 WSOP dealer’s guide that is available on a public-facing website (WSOP, 2013). However, anecdotally, it appears that a wash may also occur before any hand at the dealer’s discretion, such as what can be observed in the 2018 European Poker Tour Monte Carlo Event (PokerStars, 2018).

Unlike the riffle shuffle, the wash shuffle has remained largely unstudied in the literature. The only attempt to probabilistically model the wash shuffle that we have been able to find is in an unpublished manuscript (White, 2019). In this manuscript, three probabilistic models are proposed, in order of increasing complexity:

* Shuffling Process A: Cards begin by occupying spaces 1 through *n*, where *n* is the total number of cards. At each step, a random card is chosen, and either moves one step to the left, one step to the right, or stays in place, with probabilities of ¼, ¼ and ½ respectively. If a card moves to an occupied position, it is inserted into the pile of cards currently there at a random position. This is repeated for some specified number of steps, and then all piles are gathered together in order.
* Shuffling Process B: Cards again begin by occupying spaces 1 through *n*. At each step, every card moves a random number of spaces to the right according to independent geometric random variables, and then every card moves a random number of spaces to the left according to the same distribution. When cards enter an occupied position, they are merged with the cards already there by a GSR shuffle. This is repeated for some specified number of steps, and then all piles are gathered together in order.
* Shuffling Process C: This is a generalization of Shuffling Process A into multiple dimensions. For some number of dimensions *d*, cards begin by occupying arbitrary vertices in a *d-*dimensional grid. At each step, choose a random card. With probability ½, it remains at its current vertex; otherwise, with equal probability it will move to any neighboring vertex. If a card moves to an occupied position, it is merged with the cards already there at a random position. This is repeated for some specified number of steps, and then all piles are gathered together in order.

By the author’s own admission, each of these are quite crude models of an actual wash shuffle. Their purpose of existence was to be starting points for demonstrating particular mathematical properties such as theoretical mixing times and strong stationary times. Conversely, in the present work, our aim is to demonstrate the resulting impact of the wash shuffle on the play of Texas Hold’em via Monte Carlo simulation, and thus we are less restricted to simple models and are only restricted by the complexity of simulation code that we wish to write. Thus, we will not directly consider any of these models, and instead formulate our own.

**Methodology**

**Simulating the Wash Shuffle**

*Initial setup*

Consider that a dealer’s wash movement initially spreads a single vector of cards into two overlapping circles. These will be the left and right circles, corresponding to which hand is washing them. The left hand moves the first quarter of cards upward and then the second quarter of cards downward, thus completing the left circle. The right hand moves the fourth quarter of cards upward and then the third quarter of cards downward, thus completing the right circle.

1A

1B

2A

2B

3A

3B

4A

4B

1B

1A

2A

2B

3B

3A

4A

4B

Figure 1 Initial setup of wash shuffle

The above diagram presents the initial setup. The second diagram in Fig. 1 depicts how a wash can be thought of as a matrix of four columns. The first and third columns are cycled through by the left hand, and the second and fourth columns are cycled through by the right hand. Each completion of a circular movement by a single hand will be referred to as a single wash cycle. Next, the functions of modeling a wash shuffle will be explained.

*Card shift*

As one hand moves along, the cards in a column reach the end of their column and move to the beginning of the next column. This motion continues until the wash cycles are completed. Recall that approximately four wash cycles are completed by each hand. The number of elements by which a single wash cycle moves its cards must be specified. Each time a wash cycle moves through its columns, a random variable is generated by a truncated Poisson (TPoisson) distribution with minimum one card and maximum 52 cards. This distribution is important later on for generating a random value for how many cards a wash will swap from one column to another. It must be truncated to keep the maximum number from being infinite. The function for a truncated Poisson random variable is as such.

For both the left-hand and right-hand cycles, the number of elements by which cards shift is .

*Card switch*

With each cycle, there is a nonzero probability of a single card or group of cards switching from one column to an adjacent one. Because both hand motions overlap each other at two points, and the innermost columns move opposite to one another, card switches happen in larger groups of cards between columns two and three which are the inner columns. The number of cards for an inner and outer switch are and respectively.

Suppose that *A* and *B* equal the lengths of columns two and three respectively. A switch function generates a random sample between with probability distribution,

No card switch is one-for-one; the switch simply inserts the card or cards into the opposite column. Therefore, the longer column has a greater probability of losing its cards to a switch. The shorter column then gets longer. The index of the first card in the switch group must also be specified as, , where *L* is the length of the column from which cards are leaving. We specify *I* as such to maintain randomness while keeping the code supporting the model from running into indexing errors. If *Y* + *I* is greater than the length of its column, then *I* becomes the index of the last card in the switch group. Regardless of which cards are chosen to leave their column, they will be inserted into the same indices of the receiving columns, and the rest of the cards in the receiving column will slide down to make room. Thus, it is not assumed that each column will be of the same length.

*Gathering cards*

Once approximately four cycles have been performed by each hand, the cards are then gathered back into a single deck. While cards remain in at least one column, we generate *N*, *I*, and *L* variables such that with a mean of six, *I* randomly picks a remaining column, and *L* is the length of the chosen column. From column *I*, we move the *N* cards at the bottom of the column and place them at the top of the deck. If then we take the remaining cards in the column instead. *N*, *I*, and *L*are generated after each time a grouping of cards is moved into the deck. This process is completed once all columns are empty and the deck is of length 52.

**Pseudocode breakdown**

This section will provide a full breakdown of all functions, and the wash shuffle model overall, in pseudocode.

*Wash function:*

Input: deck = 1:52, cycles = integer for number of wash cycles

deck split into columns at ***Binom(n=52, p=0.25)***, ***Binom(n=52, p=0.5)***, ***Binom(n=52, p=0.75)***

Initialize: left\_vector = first column of card deck, reversed third column of card deck

Initialize: right\_vector = reversed fourth column of card deck, second column of card deck

FOR cycle in number of cycles:

n = amount of circular roll for left\_vector, ***TPois(lambda=3)*** distributed

CALL shift function on left\_vector for n spaces

FOR ***TPois(lambda=1.5)*** number of times:

choose slice type: slice\_inner 70% likely, slice\_outer 30% likely

IF slice\_inner chosen:

CALL slice\_inner function

ELSE IF slice\_outer chosen:

CALL slice\_outer function

END IF

n = amount of circular roll for right\_vector, ***TPois(lambda=3)*** distributed

CALL shift function on right\_vector for n spaces

FOR ***TPois(lambda=1.5)*** number of times:

choose slice type: slice\_inner 70% likely, slice\_outer 30% likely

IF slice\_inner chosen:

CALL slice\_inner function

ELSE IF slice\_outer chosen:

CALL slice\_outer function

END IF

END FOR

CALL gather function to bring vector back into single deck

RETURN deck

*Shift function:*

Input: vector, n = integer for shift spaces

Initialize: len = vector length

Initialize: new vector = vector[(n + 1):len], vector[1:n]

RETURN new vector

*Truncated Poisson (TPois) function:*

Input: lambda, max = integer for maximum possible value

Initialize: vector = empty vector with length = max

FOR x in 1 through max:

vector[x] = ***Poisson(lambda=lambda, x=x)***

END FOR

denominator = sum(vector)

vector = vector / denominator

output = sample from 1:max with vector as probability distribution

RETURN output

*Slice\_inner function:*

Input: left\_vector, right\_vector

Initialize: left\_vector\_inner = left\_vector[halfway:end of left\_vector]

Initialize: right\_vector\_inner = right\_vector[halfway:end of right\_vector]

Initialize: len1 = length of left\_vector\_inner

Initialize: len2 = length of right\_vector\_inner

Initialize: p = len1 / (len1 + len2), probability of slicing from left\_vector\_inner

Initialize: condition = choose left\_vector\_inner or right\_vector\_inner based on probability p

*Slice\_outer function:*

pseudocode

*Gather function:*

Input: left\_vector, right\_vector

Initialize: column1 = second half of left\_vector

Initialize: column2 = first half of left\_vector

Initialize: column3 = second half of right\_vector

Initialize: column4 = first half of right\_vector

Initialize: column\_lengths = static list of each column length

Initialize: deck = empty deck, counter = 52, columns = 4

WHILE count not equal to zero:

i = ***Binom(n=columns, p=0.5)***, more likely to sample from column2 and column3

length = length of ith column (note this is the local column length for current while loop)

n = ***Binom(n=column\_length at i*** (static column length outside while loop)***, p=0.5)*** + 1

IF n = 0:

NEXT

ELSE IF length > 0: (if cards exist in ith column)

IF length > n:

slice = last n cards in ith column

deck = add sliced cards to beginning of deck

ith column = delete sliced cards

counter = decrement by n

ELSE IF length = n: (if whole remainder of ith column will be sliced)

slice = ith column

deck = add sliced cards to beginning of deck

ith column = DELETE

counter = decrement by n cards

columns = decrement by 1 (one less column to gather from)

ELSE IF length < n: (if n is greater than length of ith column)

slice = ith column

deck = add sliced cards to beginning of deck

ith column = DELETE

counter = decrement by n cards

columns = decrement by 1 (one less column to gather from)

END IF

ELSE IF ith column = 0: (just to make sure it skips empty column)

NEXT

END IF

RETURN deck

**A single wash shuffle and a Monte Carlo simulation**

The process for a singular wash shuffle follows. The deck of cards is initialized from one to 52. Then the number of cycles is set to four, for each hand, so a total of eight. For each cycle, the cards shift by a truncated Poisson random variable. Then a 50-50 choice is made between performing an outer or inner shuffle. Note that an outer shuffle on the left hand will switch between columns one and two while a shuffle on the right hand will switch between columns three and four. If the left cycle was previously executed, then the right cycle goes next. After all cycles are completed, the columns of cards are gathered into a deck.

This process is ultimately fed into a Monte Carlo simulation where, for 100,000 repetitions, a new deck is initialized, and the wash shuffle is executed as described. Throughout the Monte Carlo sim, we take the newly shuffled deck and record the previous locations of cards 14, 15, 16, 18, and 20 because these will be the community cards in a game of NLHE.

**References**

Aldous, D., & Diaconis, P. (1986). Shuffling cards and stopping times. *The American Mathematical Monthly,* *93*(5), 333. doi:10.2307/2323590

Black, J. (2020). How to shuffle cards professionally. *Jack Black Casino Dealer School.* Retrieved from <https://www.youtube.com/watch?v=svi7Jry-KQI&t=344s>

Gilbert, E. (1955). Theory of shuffling. Technical memorandum, Bell Laboratories.

Merz, D. W., & Chi, P. B. (2022). Knowing When to Fold'em: A Monte Carlo Exploration of Card Shuffling and How Poker Players Can Gain an Advantage. UNLV Gaming Research & Review Journal, 26(1), 7.

PokerGO. (2017). 2017 WSOP Main Event Day 1. Retrieved from <https://www.youtube.com/watch?v=zdTlTqqtcIk&t=420s>

PokerGO. (2022). 2022 WSOP Main Event Day 6. Retrieved from https://www.youtube.com/watch?v=p6\_t9ns9rUw&t=2h4m9s

PokerStars. (2018). Main Event Day 2 - EPT Monte Carlo 2018. Retrieved from <https://www.youtube.com/watch?v=WuNZEccZOsg&t=1917s>

Reeds, J. (1981). Unpublished manuscript.

White, G. (2019). A variation of strong stationary times for random walks with partial symmetries. arXiv preprint arXiv:1910.02389.

WSOP. (2013). 2013 World Series of Poker official dealer guide. Retrieved from <https://www.wsop.com/2013/2013_WSOP_Dealer_Guide.pdf>