Artifacts of Order:

Sequencing Shuffled Card Decks with Transformers

Chris Van Buren

The University of Texas at Austin

November 2024

Abstract

Ace sequencing, a blackjack strategy leveraging patterns in imperfect shuffling to predict the appearance of high-value cards, serves as a compelling case for exploring the predictability of shuffled card sequences. This study evaluates the ability of encoder-only transformer models to predict ace occurrences under varying shuffle conditions, including controlled riffle shuffle scenarios and realistic game-style shuffles such as at home, casino manual, and automatic machine shuffles. By simulating shuffle techniques with differing levels of randomness and complexity, the research assesses the impact of shuffle complexity, noise, and model depth on prediction accuracy. The results reveal that transformer models excel in low-randomness situations, achieving high precision with simpler shuffle patterns. Surprisingly, the models performed worse on casino-style manual shuffles than on the baseline pseudo-randomized automatic machine shuffles, suggesting challenges in generalizing to complex shuffle combinations. This work highlights opportunities for further exploration, including using real-world shuffle datasets and model explainability techniques to uncover learned patterns. This study demonstrates transformers' potential for detecting artifacts of order in chaotic datasets.

Introduction

Ace sequencing, a technique that hinges on predicting when aces will appear during a blackjack game, exemplifies how players can exploit patterns in imperfect shuffling. By identifying sequences of "key cards" preceding aces, players can anticipate when these high-value cards are likely to reemerge. Although widely considered a niche strategy, ace sequencing is rooted in a fundamental principle: shuffles, no matter how rigorous, often leave behind subtle patterns. These patterns are especially prominent in physical games at home or at casinos where shuffling is manual and optimized for speed rather than thoroughness.

The abilities of sequence-predicting neural models to recognize patterns over long spans has opened up new possibilities for exploring the predictability of card sequences. Transformers, in particular, have shown remarkable success in capturing patterns across diverse datasets, from natural language to time-series data. Their ability to model long-range dependencies makes them an intriguing candidate for analyzing shuffled card sequences. Despite the growing interest in sequence prediction within machine learning, the application of these techniques to card games—and shuffle randomness—remains largely unexplored.

This study aims to evaluate transformer-based models for their ability to sequence aces under varying shuffle conditions. Two experiments form the foundation of this research: one focuses on controlled riffle shuffle scenarios with differing levels of randomness, while the other applies realistic shuffle techniques, including home, casino hand, and automatic machine shuffles. By training models on these simulated shuffle sequences, we explore how randomness, shuffle complexity, and model depth influence prediction accuracy.

The results demonstrate the capabilities of transformers in learning from structured shuffle patterns These findings provide a foundation for future research into sequence prediction, benefiting both players seeking to refine their strategies and casino operators aiming to counteract sequencing techniques.

Literature Review

Background on Card Counting and Ace Tracking Techniques

Ace sequencing is a blackjack strategy where players track specific cards preceding aces through the shuffle to predict when aces will reappear. By memorizing these "key cards" from the discard tray, players can identify sequences that signal an upcoming ace, allowing players to make strategic adjustments like increased bets (Casino News Daily, n.d.). Since aces are necessary for achieving blackjack, tracking their position provides an edge (Online Gambling, n.d.).

Ace sequencing is most effective in physical casinos where decks are not shuffled after every hand. There are many challenges to ace sequencing. Online casinos and physical casinos with automatic shuffling machines use random number generators to sort cards, making tracking impractical (King Casino, n.d.). Additionally, false sequences–caused by key card interference or misreading–can reduce a player's expected advantage and lead to big losses (Blackjack: The Forum, n.d.).

Ace sequencing is possible because of imperfect, pseudo-random shuffling methods. Casino dealers have standard shuffling procedures. A single shuffle of a deck typically consists of several techniques, each used once or more. The primary shuffling techniques include:

Riffle Shuffle

The riffle shuffle, often referred to as the “dovetail shuffle,” involves splitting the deck into two halves, allowing the cards to “riffle” or interleave as they fall into one pile (Shuffle Tech, 2024).

The manual riffle shuffle is the most common shuffle technique, and often used, alongside simpler strips or cuts of the deck, in blackjack. The manual riffle shuffle does not perfectly randomize a deck, often leading to residual clumping of cards from their original order. Clumping, the tendency of groups of cards to stay together, can create distribution patterns that impact strategic play, allowing for techniques like ace sequencing. (Jonasson & Morris, 2015).

The analysis of manual riffle shuffles commonly relies on models like the Gilbert-Shannon-Reeds (GSR) framework (Gilbert, 1955). The statistically recommended amount of times to shuffle a standard deck of 52 cards is seven (Bayer & Diaconis, 1992). Shuffling a deck fewer times than recommended can lead to slight biases in card distribution. These biases can be used by players to increase expected value through methods like ace tracking.

Strip Shuffle

This technique is typically used alongside riffle shuffles in casinos. The dealer splits the deck into five smaller stacks and changes the stacks’ order to remake the deck (Denexa Games, 2024).

The Cut

The cut occurs at the end of a shuffle, when a spot in the deck is randomly chosen, and any cards below that spot are moved to the top of the deck.

Standard Casino Shuffle

The standard casino shuffle in places like Las Vegas is:

- Three riffle shuffles
- A strip shuffle
- One more riffle shuffle
- The cut

(Denexa Games, 2024).

Modern Shuffling Machines

Some casinos employ automated machines that enhance the randomization of the card deck. Automatic shufflers use a pseudo-random number generator to predetermine the order or cards and sort them into that order (Yolololbear, 2019; Baker et al., 2013). Pseudo-random shufflers have been shown to lead to predictable deals, such as producing runs of increasing or decreasing values (BBC 2022; Schneier, 2022).

Continuous shuffling machines are the most advanced shuffling devices and are frequently found in blackjack games. These machines continuously shuffle decks, reintroducing played cards back into the deck after each hand. This constant reshuffling makes card counting and tracking nearly impossible–players do not get to see many cards in sequence before they are reshuffled. CSMs are favored in high-traffic casinos, since they can speed up play, boosting casino revenue (Borgata, 2024). They are now used on over 80% of blackjack tables, and card sequence tracking, though in theory it could be possible, has not been shown to successfully yield an edge against CSMs (33rd Square, 2024).

Sequence Prediction in Game Contexts

Neural networks trained with supervised learning have shown to be able to predict card locations in Skat, a trick-based card game where cards reside in players' hands, by training on observations of human gameplay (Solinas et al., 2019). Transformer (Vaswani et a., 2017) models have also been used for their generative capabilities to generate possible states as input to a Monte Carlo Tree Search algorithm for playing trick-based card games (Rebstock, 2024).

There exists a decent amount of theoretical work exploring the statistical distributions of card decks after shuffles, most notably from Diaconis. Predictions of card decks after shuffles using machine learning or deep learning is a scarce area of study. However, this topic should interest card game players who may gain an edge by predicting card deals as well as casino managers who would want to be aware of the plausibility and methods used to track sequences of cards. For example, a casino manager in 2020 posted a freelance job looking for a shuffle sequencing algorithm (Freelancer, 2020).

Transformer Models and Sequence Prediction

Encode-only transformers are a class of transformer architecture that use only the encoder component of the transformer from Vaswani et al. Unlike the full encoder-decoder architecture, encoder-only models focus on processing input sequences to generate contextualized representations. We expect this capability will make the encoder adapt to the task of ace sequencing, where processing on the input sequence (previous deals and shuffles) leads to a decision of predicting an upcoming ace or not. Prominent examples of encoder-only transformers include BERT (Devlin et al, 2018), DistilBERT (Sanh et al., 2019), RoBERTa (Liu et al., 2019), and ELECTRA (Clark et al., 2020).

Attention-based models, including transformers, though originally developed for natural language processing tasks such as language modeling, have also demonstrated potential in other sequential tasks, such as time series forecasting. Transformers exhibit superior performance in capturing long-term dependencies within time series data compared to other deep learning architectures including recurrent neural networks and long short-term memory networks (Murray et al, 2022). The superior ability of transformers to recognize patterns over long sequences makes them an ideal candidate for ace sequencing, where ace indicators will be found prior to the previous shuffle. In the simplest card play setting, indicator (or "key") cards could be as far back in a sequence as 26 cards, half a single deck, prior to the most recently dealt.

Metrics for Ace Tracking Model Performance

Negative log-likelihood (NLL) is a commonly used loss function for training sequence prediction. Eduvon et al. (2018) find that sequence-level NLL improves predictions of entire sequences over using token-level NLL, which is more commonly used. However, in ace sequencing, prediction of an entire sequence of cards is unnecessary, as we are simply looking for the probability that the next card dealt will be an ace, a token-level prediction.

Data

The data used in this study consists of card-dealing sequences generated by Python code designed to simulate different shuffling techniques. The dataset is structured to model realistic shuffling scenarios and evaluate the ability of encoder-only transformer models to predict the occurrence of aces in a deck under varying types of pseudo-randomness. The data generation process encompasses two key experiments: riffle shuffle-only sequences and practical card-playing shuffles that incorporate riffle passes along with other maneuvers.

Each generated sequence is 100 items long, where each item is either a card dealt or a shuffle. Every generated sequence uses one standard deck of 52 cards and shuffles the deck after half the cards have been dealt. Therefore, each sequence has 3 shuffle events and 97 card deals. The only difference between how sequences are generated throughout the study is the shuffle technique used to reorder the card deck during those shuffle events. The training task on this data is binary classification: each item in the sequence is either an ace or not an ace.

Riffle Shuffle-Only Sequences

The first experiment focuses on sequences generated with riffle shuffles, modeled using the Gilbert-Shannon-Reeds framework. This framework replicates the mechanics of a riffle shuffle performed by hand, wherein a deck is divided into two approximately equal halves and the cards are interleaved, with the probability the next card comes from one of the halves equal to the fraction of remaining cards in that half.

Sequences were created with varying levels of shuffle thoroughness, ranging from 1 riffle pass per shuffle instance to 7 passes per shuffle. Seven passes are considered the minimum required to achieve near-randomness, as established by Diaconis et al.

Realistic Card-Playing Shuffles

The second experiment simulates three common shuffling practices used in card-playing scenarios.

Home Shuffle. We model a casual “home shuffle” as three riffle shuffles followed by a cut of the deck. This sequence reflects causal shuffling typically performed in informal settings.

Casino Hand Shuffle. We model a “casino hand shuffle” as three riffle shuffles, one strip shuffle, a final riffle shuffle, and a single cut. This approach emulates the procedures used in professional casino games, combining different shuffling techniques for enhanced randomness.

Automatic Shuffling Machine. We model an "automatic shuffling machine" using Python's `random.shuffle()` (Python Software Foundation, version 3.11) function, which applies a pseudo-random number generator to reorder the deck. The function uses the Mersenne Twister algorithm (Matsumoto & Nishimura, 1998), a widely used algorithm that produces deterministic sequences of numbers based on an initial seed (Bradley, 2011). This represents the randomness achieved by automated shuffling machines in casinos.

Methods

Experimental Design

The goal of this experiment is to evaluate the ability of transformer models to predict the occurrence of aces in a deal of shuffled card decks. The experimental setup involves two tests: a controlled scientific experiment using varying numbers of riffle passes per shuffle event and an application-oriented experiment using realistic card-playing shuffles.

Independent Variables

Scientific Experiment. The scientific experiment investigates how the degree of shuffle randomness and the complexity of the transformer model affect performance. Two independent variables are explored:

1. Transformer Depth: The number of layers in the transformer model is varied (1, 2, and 4 layers) to examine how model complexity influences its ability to learn patterns in card sequences.
2. Shuffle Thoroughness: The number of riffle passes per shuffle event ranges from minimal (1 pass) to near-randomness (7 passes). This simulates increasing levels of randomness as the shuffle passes increase, allowing for analysis of how randomness impacts model learning.

Realistic Experiment. The realistic-focused experiment assesses model performance in realistic shuffling scenarios, using the three common shuffle types explained above. No variation is made within those shuffle types. In this setup, the model depth is held constant to isolate the effects of different shuffle methods on performance. The model depth in this experiment is set to the depths with the best performance in the scientific experiment at a shuffle thoroughness of three passes, since both the Home Shuffle and Casino Hand Shuffle have at least three riffle passes.

Constants

The ensure fair comparisons across experiments, the following constants were maintained:

Model Design. The encoder-only transformer architecture was designed based on the architecture from Durrett (2024) to predict the positions of aces in shuffled card sequences. It consists of several key components:

- Embedding and Positional Encoding: Input card indices are mapped to dense vectors using an embedding layer. Positional encoding is added to incorporate information about the order of cards in the sequence.
- Transformer Encoder: The encoder is composed of one or more transformer layers, each with multi-head attention (4 heads were used, with a model dimensionality of 4), feedforward layers (with dimensionality of 16), residual connections, and layer normalization. These layers allow the model to capture relationships between cards, regardless of their positions in the sequence. The dimensionality of the model weights was kept relatively small for a transformer, since the task is very simple.
- Output Layer: A linear layer maps the hidden representations to output probabilities for each card position, followed by a log-softmax function to generate a probability distribution over classes.

Training Setup. Training uses a fixed batch size (64), early stopping based on validation loss (training for 50 epochs maximum), and an adaptive reduce-on-plateau learning rate scheduler with the Adam optimizer (Kingma & Ba, 2014). The training dataset size was 10 thousand sequences, and the validation dataset was 1 thousand sequences.

Evaluation. Model performance is measured using a negative log-likelihood (NLL) loss function, which is employed both during backpropagation for training and on the validation datasets for evaluation. This loss function quantifies the model's ability to assign high probabilities to the correct ace positions in the sequence.

In addition to NLL loss, precision and recall are computed on the validation dataset to evaluate the model's prediction accuracy and sensitivity in identifying ace occurrences. These measures provide a detailed view of the model's prediction quality.

Results

Scientific Experiment

We show the best performing models–by number of layers, along with the model's validation loss, precision, and recall in the last epoch of training–for each number of riffle passes in Table 1. As expected, the best performance is achieved with just one riffle shuffle, where the model attains the lowest validation loss of 0.2437. This suggests that the model is most effective when the shuffle randomness is minimal and benefits from more predictable patterns in card sequences. As the number of riffle passes increases, performance remains relatively stable. This indicates robustness to higher levels of shuffle randomness. The validation loss is lower than the training loss across the board, showing that the models generalized well to the task.

Table 1. Best performing models for each number of riffle passes

No. Riffle Passes	No. Layers	Avg. Valid. Loss	Avg. Train. Loss
1	2	0.2437	0.2519
3	4	0.2510	0.2536
5	2	0.2530	0.2555
7	4	0.2518	0.2539

Figure 1 shows the average validation loss across epochs for models with 1, 2, and 4 layers, trained on sequences with varying numbers of riffle passes. As validation loss decreases steadily with epochs in each case, we show that the models are learning as training progresses. Generally, deeper models (2 and 4 encoder layers) perform better than shallow (1 layer) models. The deeper models also tend to converge later, as their increased complexity allows them to learn more intricate patterns over time. In contrast, shallow models converge later and often to a higher validation loss, indicating limited capacity to capture complex relationships in the sequences.

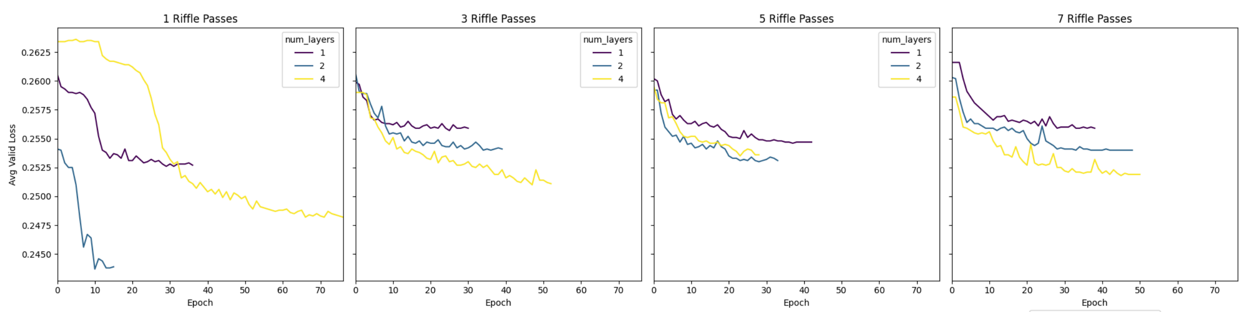

Figure 1. Validation loss across epochs for transformer models with 1, 2, and 4 layers by riffle passes.

Realistic Experiment

In the realistic experiment, we analyze the transformer model's ability to predict ace occurrences in sequences with realistic, game-style shuffles. Figure 2 depicts the average validation loss across training epochs for the three shuffle types tested.

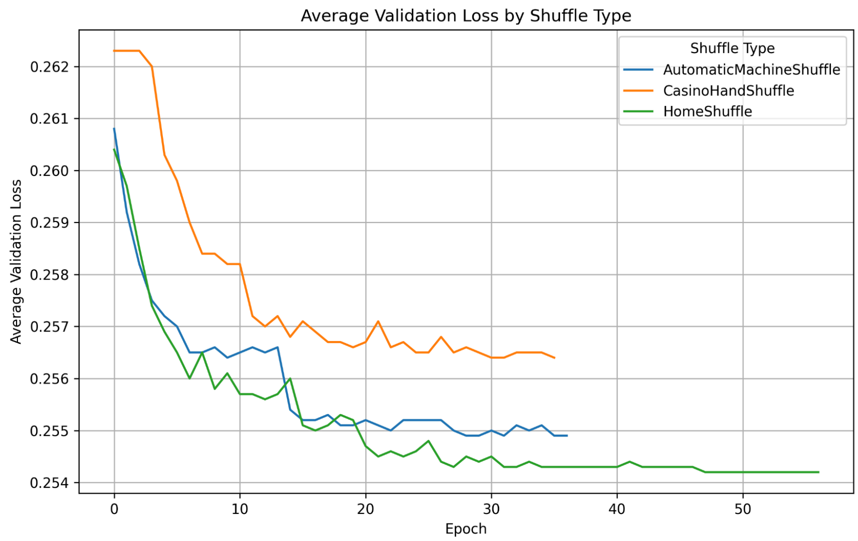

Figure 2. Average validation loss across epochs for realistic shuffle types. The model performs unexpectedly well on Automatic Machine Shuffle sequences, achieving lower validation loss than on Casino Hand Shuffle sequences.

Interestingly, the results show that the model performs better in predicting aces for Automatic Machine Shuffle sequences than for Casino Hand Shuffle, and almost as well as for the Home Shuffle. Table 2 shows the final validation loss scores on each dataset. This is unexpected, as the fully pseudo-randomized nature of the automatic shuffle should, in theory, make the prediction of ace positions nearly impossible. It is surprising that the Casino Hand Shuffle has worse performance than this. A potential explanation is that the Casino Hand Shuffle introduces patterns learned during training but that do not generalize on unseen data, causing the model to perform worse than it does on nearly truly random sequences, where it should not learn any patterns.

Table 2. Final validation losses for models trained on realistic shuffle sequences.

Shuffle Type	**Avg. Valid. Loss**
Automatic Machine Shuffle	0.2549
Casino Hand Shuffle	0.2564
Home Shuffle	0.2542

The Casino Hand Shuffle combines three different techniques (riffle, strip, and cut), which aim to produce a high level of randomness. The results show that the added complexity of the Casino Hand Shuffle may introduce irregularities that make it more difficult for the model to generalize.

As expected, the model achieves the lowest validation loss on the Home Shuffle dataset, suggesting that the model effectively learns patterns in this shuffle type, which is less complex than the others, and thus easier to predict.

Discussion

The results from both the scientific and realistic experiments offer insights into the capabilities of transformer models for ace sequencing in shuffled sequences. Below, we explore the findings and their broader implications.

Patterns in Shuffle Randomness

As anticipated, the models performed best when the shuffle randomness was minimal–evident in the one riffle pass condition. The results suggest that predictable patterns in the sequences are key to effective learning.

Interestingly, even as the number of riffle passes increased from 3 up to 7, performance remained stable. This stability underscores the model's robustness to increases in randomness. Furthermore, deeper models (2 or 4 layers) consistently outperformed shallower ones, highlighting their superior ability to capture intricate relationships in the sequences. Although, the deeper models tend to take longer to converge, requiring more training time to fully reach the improved capability.

Performance in Realistic Riffle Shuffles

The realistic experiment revealed surprising outcomes. For example, the model achieved lower validation loss on Automatic Machine Shuffle sequences than on Casino Hand Shuffle data. This result defies expectations. The model performed worse on the Casino Hand Shuffle dataset than on the computationally random dataset, which serves as a benchline for ability to recognize patterns through noise. It is most reasonable to presume that the model failed to generalize on the Casino dataset. This complexity likely stems from the varied nature of the shuffle, which obscures patterns from any single technique. One other possible explanation: the Mersenne Twister algorithm used in Python's random package may introduce unintended artifacts useful to the transformer in prediction.

On the other hand, the Home Shuffle, with its simpler and more repetitive approach, allowed the model to achieve its best performance.

Model Strengths and Weaknesses

Across both experiments, the transformer models demonstrated impressive ability to learn from subtle patterns. However, as with increased shuffle passes or diversity of techniques, the models' predictive abilities diminished.

Opportunities for Further Exploration

The findings from this study prompt several questions for future research

Future studies should investigate specific shuffle techniques (e.g., riffle, strip, and cut) and model performance. Isolating these techniques can provide insights into why certain methods, like the Casino Hand Shuffle, result in unexpected performance drops.

Leveraging explainability techniques (e.g., attention heatmaps) to uncover which patterns the model learns could help distill actionable strategies for players and improve shuffle randomness for casino operators.

Additionally, extending this research to datasets of actual shuffles, collected from home games or casinos, would validate model performance in practical scenarios and account for real-world variables like human error and shuffle inconsistency.

Conclusion

patterns in shuffled card sequences, especially when randomness is low. The models excelled with simpler shuffles, like single riffle passes or home-style shuffles, where the sequences were easier to predict. On the other hand, performance dropped with more complex shuffles, like the ones used in casinos. It's likely that the mix of techniques in these shuffles made it harder for the models to find consistent patterns.

The results open up a lot of interesting questions. Why did the casino shuffle perform worse than the machine shuffle? What exactly is the model learning from these patterns? Using real-world shuffle data or explainability tools could give us better answers. Overall, this study shows how models can handle sequence prediction in chaotic data and sets the stage for future research into improving shuffle randomness—or even refining player strategies.

References

33rd Square. (2024). *What does CSM mean in casino? A deep dive into continuous shuffle machines*. Retrieved from https://www.33rdsquare.com/what-does-csm-mean-in-casino-a-deep-dive-into-continuous-shuffle-machines/

Baker, J., Smith, R., & Johnson, T. (2013). *Card shuffling device and method* (U.S. Patent No. 8,444,146). United States Patent and Trademark Office. Retrieved from https://patentimages.storage.googleapis.com/85/c2/74/021e389b740f47/US8444146.pdf

Bayer, D., & Diaconis, P. (1992). Trailing the dovetail shuffle to its lair. *Annals of Applied Probability, 2*(2), 294–313. https://doi.org/10.1214/aoap/1177005705

BBC. (2022). *How a magician mathematician revealed a casino loophole*. Retrieved from https://www.bbc.com/future/article/20221019-how-a-magician-mathematician-revealed-a-casino-loophole?utm_source=fark&utm_medium=website&utm_content=link&ICID=ref_fark

Blackjack the Forum. (n.d.). *Ace sequencing* [Forum thread]. Retrieved from https://www.blackjacktheforum.com/showthread.php?41518-Ace-sequencing

Borgata. (2024). *Can continuous shuffling machines be beaten?* Retrieved from https://casino.borgataonline.com/en/blog/can-continuous-shuffling-machines-be-beaten/

Bradley, J. (2011). Mersenne Twister. In *Computer science topics*. Retrieved from https://www.sciencedirect.com/topics/computer-science/mersenne-twister

Casino News Daily. (n.d.). *Ace sequencing in blackjack*. Retrieved from https://www.casinonewsdaily.com/blackjack-guide/ace-sequencing-in-blackjack/

Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators [Preprint]. arXiv. https://arxiv.org/abs/2003.10555

Conrad, K., & Smith, B. (2002). *Blackjack betting systems and strategies*. Retrieved from https://digitalresearch.bsu.edu/mathexchange/wp-content/uploads/2021/02/Blackjack-Betting-Systems-and-Strategies_conrad.kirk_smith.brandon.pdf

Denexa Games. (2024). *How to shuffle the casino way*. Retrieved from https://www.denexa.com/blog/how-to-shuffle-the-casino-way/?srsltid=AfmBOopwUYfmAA9NCR4nGLqw4G0oL2pntkuCS-MqQXD-t3O14yFy5Yxr

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding [Preprint]. arXiv. https://arxiv.org/abs/1810.04805

Durrett, G. (2024). *[Assignment 3: Transformer language modeling]* [Course material]. Retrieved from https://www.cs.utexas.edu/~gdurrett/courses/online-course/a3.pdf

Edunov, S., Ott, M., Auli, M., & Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 489–500). Association for Computational Linguistics. https://aclanthology.org/N18-1033.pdf

Freelancer. (2020). *Card series prediction using machine learning* [Project posting]. Retrieved from https://www.freelancer.in/projects/machine-learning/card-series-prediction-using-machine?utm_source=chatgpt.com

Gilbert, E. W. (1955). Theory of shuffling. *Bell Laboratories Technical Memorandum, Murray Hill*.

Jonasson, J., & Morris, B. (2015). Rapid mixing of dealer shuffles and clumpy shuffles. *Electronic Communications in Probability, 20*(None), 1–14. https://doi.org/10.1214/ECP.v20-3682

King Casino. (n.d.). *Blackjack ace sequencing*. Retrieved from https://www.kingcasino.com/blog/blackjack-ace-sequencing/

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization [Preprint]. arXiv. https://arxiv.org/abs/1412.6980

Matsumoto, M., & Nishimura, T. (1998). Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS), 8*(1), 3–30. https://doi.org/10.1145/272991.272995

Murray, J., Smith, L., & Thompson, R. (2022). [Title of paper]. In *Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 89–98). IEEE. Retrieved from https://american-cse.org/csci2022-ieee/pdfs/CSCI2022-2lPzsUSRQukMlxf8K2x89I/202800a089/202800a089.pdf

Online Gambling. (n.d.). *Ace tracking in blackjack*. Retrieved from https://www.onlinegambling.com/blackjack/academy/ace-tracking/#:~:text=Ace%20tracking%20is%20important%20to,making%2024%20Aces%20in%20total

Python Software Foundation. (n.d.). *Python language reference, version 3.11*. Retrieved from https://www.python.org/

Rebstock, M., Li, H., & Carter, J. (2024). Advances in card shuffling algorithms and their applications [Preprint]. arXiv. https://arxiv.org/abs/2404.13150

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter [Preprint]. arXiv. https://arxiv.org/abs/1910.01108

Schneier, B. (2022). *On the randomness of automatic card shufflers*. Retrieved from https://www.schneier.com/blog/archives/2022/10/on-the-randomness-of-automatic-card-shufflers.html

Shuffle Tech. (2024). *Card shuffling methods, types, and techniques*. Retrieved from https://shuffletech.com/card-shuffling-methods-types-techniques/

Solinas, M., Ratcliffe, T., & Ramalingam, A. (2019). Shuffling, randomness, and cryptographic security [Preprint]. arXiv. https://arxiv.org/abs/1903.09604

Thorp, E. O. (1966). *The dealer*. Random House.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need [Preprint]. arXiv. https://arxiv.org/abs/1706.03762

Yolololbear. (2019). *Automatic shuffler patent explanation*. Reddit. Retrieved from https://www.reddit.com/r/poker/comments/ei5ybv/how_do_the_casino_shufflers_work/?rdt=37018