

Correctness and Performance Charts

Peter Occil

1 Correctness and Performance Charts

This version of the document is dated 2025-01-27.

The following charts show the correctness of many of the algorithms in “**Bernoulli Factory Algorithms**¹” and show their performance in terms of the number of bits they use on average. For each algorithm, and for each of 100 λ values evenly spaced from 0.0001 to 0.9999:

- 500 runs of the algorithm were done. Then...
- The number of bits used by the runs were averaged, as were the return values of the runs (since the return value is either 0 or 1, the mean return value will be in the interval $[0, 1]$). The number of bits used included the number of bits used to produce each coin flip, assuming the coin flip procedure for λ was generated using the `Bernoulli#coin()` method in *bernoulli.py*, which produces that probability in an optimal or near-optimal way.

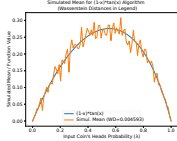
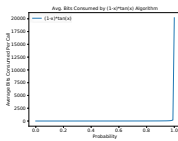
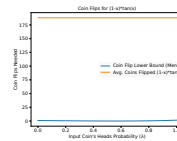
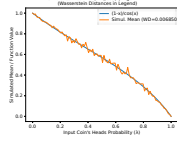
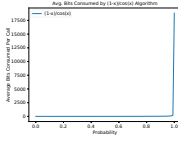
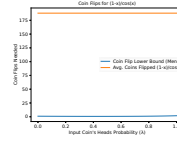
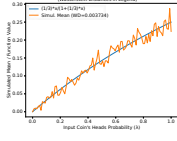
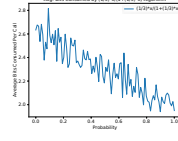
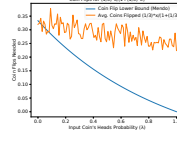
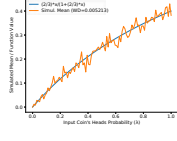
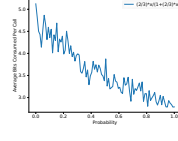
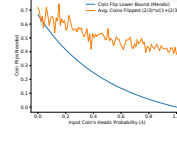
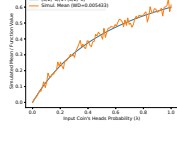
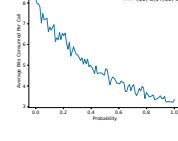
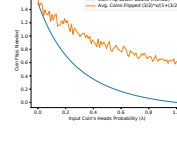
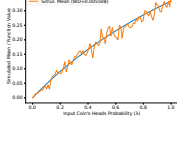
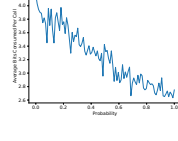
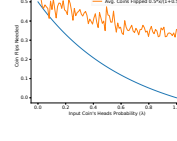
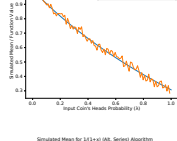
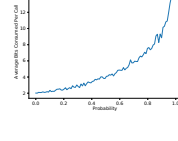
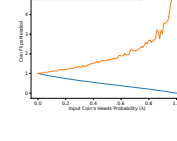
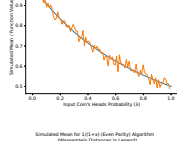
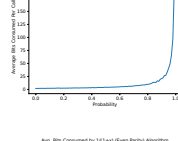
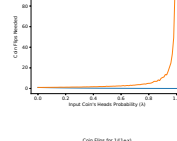
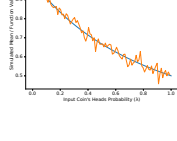
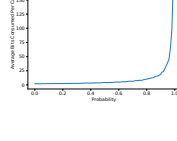
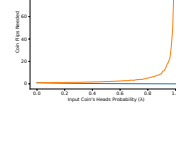
For each algorithm, if a single run was detected to use more than 5000 bits for a given λ , the entire data point for that λ was suppressed in the charts below.

In addition, for each algorithm, a chart appears showing the minimum number of input coin flips that any fast Bernoulli factory algorithm will need on average to simulate the specified function, based on work by Mendo (2019)^[^1]. Note that some functions require a growing number of coin flips as λ approaches 0 or 1. Note that for the 2014, 2016, and 2019 algorithms—

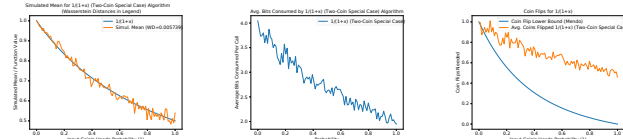
- an ϵ of $1 - (x + c) * 1.001$ was used (or 0.0001 if ϵ would be greater than 1), and
- an ϵ of $(x - c) * 0.9995$ for the subtraction variants.

Points with invalid ϵ values were suppressed. For the low-mean algorithm, an m of $\max(0.49999, xc1.02)$ was used unless noted otherwise.

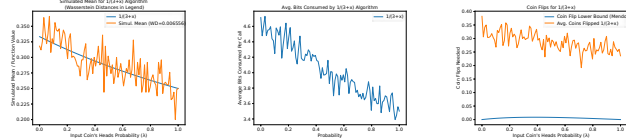
1.1 The Charts

Algorithm	Simulated Mean	Average Bits Consumed	Coin Flips
$(1-x)*\tan(x)$			
$(1-x)/\cos(x)$			
$(1/3)*x/(1+(1/3)*x)$			
$(2/3)*x/(1+(2/3)*x)$			
$(3/2)*x/(1+(3/2)*x)$			
$0.5*x/(1+0.5*x)$			
$1 - \ln(1+x)$ (Alt. Series)			
$1/(1+x)$ (Alt. Series)			
$1/(1+x)$ (Even Parity)			

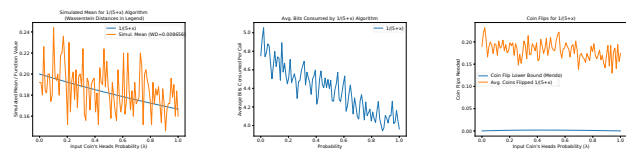
1/(1+x) (Two-Coin Special Case)



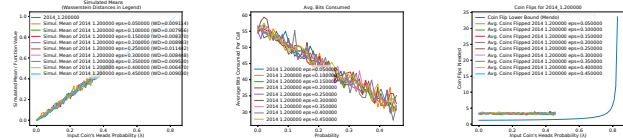
1/(3+x)



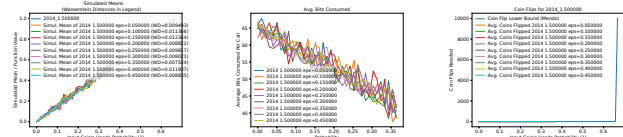
1/(5+x)



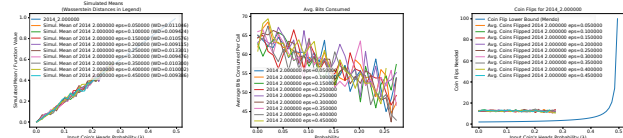
2014 1.200000
eps=0.050000



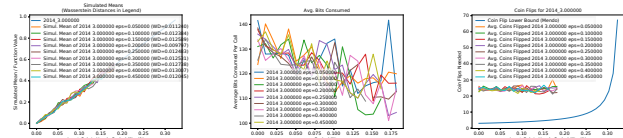
2014 1.500000
eps=0.050000



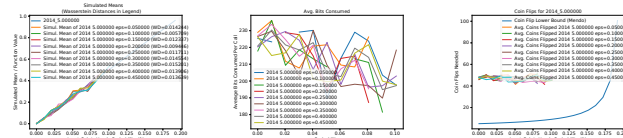
2014 2.000000
eps=0.050000



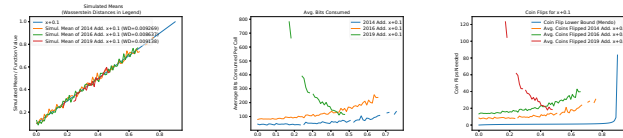
2014 3.000000
eps=0.050000



2014 5.000000
eps=0.050000



2014 Add. x+0.1



2014 Add. x+0.2

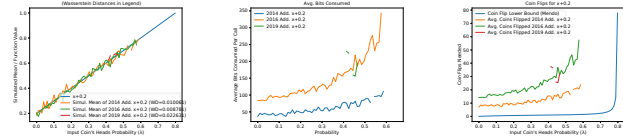


Figure 1 consists of three subplots comparing the proposed method (blue line) with the state-of-the-art (orange line). The left subplot shows the 'Empirical Mean' of the 'Spectral Power' (y-axis, 0.0 to 0.4) versus 'Frequency' (x-axis, 0.0 to 0.5). The middle subplot shows the 'Avg. Bias Coefficient' (y-axis, 0 to 250) versus 'Frequency' (x-axis, 0.0 to 0.5). The right subplot shows the 'Coeff. Bias' (y-axis, 0 to 10) versus 'Frequency' (x-axis, 0.0 to 0.5). The legend for all plots is: 'Proposed Method' (blue line), 'State-of-the-Art' (orange line), 'Avg. Bias Coefficient' (blue line), and 'Coeff. Bias' (orange line).

Figure 1 consists of three subplots illustrating the performance of the proposed algorithm. The left subplot shows the Normalized Mean Squared Error (MSE) versus Input Class Probabilities (x) for 2000 Samples per Class and 2000 Samples per Class - 10%. The middle subplot shows the Average Bit Count per Element versus Probabilities for 2004 data with 100, 1000, and 10000 elements. The right subplot shows the Cost (bits per element) versus Input Class Probabilities (x) for 2004 data with 100, 1000, and 10000 elements, along with the Average Cost for 2004 data with 10000 elements.

Figure 1 consists of three subplots comparing the proposed method (red line) with other methods (various colored lines) across different metrics.

Left Subplot: Data Normalized by Variance vs Input Core-Channel Frequency (Hz)

This plot shows the relationship between the input core-channel frequency and the data normalized by variance. The x-axis ranges from 0.0 to 0.8 Hz, and the y-axis ranges from 0.0 to 1.0. The data points for all methods (including the proposed method) follow a similar linear trend, indicating that the proposed method's normalization is consistent with other methods.

Middle Subplot: Avg. Info. Coefficient vs Probability

This plot shows the average information coefficient as a function of probability. The x-axis ranges from 0.0 to 0.7, and the y-axis ranges from 0 to 400. The proposed method (red line) shows a sharp peak at probability 0, reaching a value of approximately 400. Other methods show much lower values, generally below 100.

Right Subplot: Cost for Top-10 vs Input Core-Channel Frequency (Hz)

This plot shows the cost for the top-10 results as a function of input core-channel frequency. The x-axis ranges from 0.0 to 0.8 Hz, and the y-axis ranges from 0 to 1500. The proposed method (red line) shows a sharp peak at frequency 0.5, reaching a value of approximately 1500. Other methods show much lower costs, generally below 100.

Figure 1 consists of four subplots (a, b, c, d) showing the performance of the proposed algorithm. Subplot (a) is a Receiver Operating Characteristic (ROC) curve for the proposed algorithm, plotting True Positive Rate (TPR) on the y-axis (0.0 to 1.0) against False Positive Rate (FPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a strong performance. Subplot (b) is an Average ROC curve for the proposed algorithm, plotting Average True Positive Rate (ATPR) on the y-axis (0.0 to 1.0) against Average False Positive Rate (AFPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a strong performance. Subplot (c) is an Average ROC curve for the proposed algorithm, plotting Average True Positive Rate (ATPR) on the y-axis (0.0 to 1.0) against Average False Positive Rate (AFPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a strong performance. Subplot (d) is an Average ROC curve for the proposed algorithm, plotting Average True Positive Rate (ATPR) on the y-axis (0.0 to 1.0) against Average False Positive Rate (AFPR) on the x-axis (0.0 to 1.0). The curve is a solid blue line, showing a strong performance.

[illegible]

Figure 1 consists of three subplots labeled (a), (b), and (c), each showing the performance of various models under different weather conditions and input CAGR values.

Subplot (a): Dry Weather (CAGR) vs. Input CAGR's mean probability (X)

The y-axis represents Dry Weather (CAGR) in %/yr, ranging from 0 to 1.5. The x-axis represents Input CAGR's mean probability (X), ranging from 0.0 to 1.0. The legend includes:

- Prop. (blue line)
- Other (grey line)
- Prop. of 2014 (red line)
- Prop. of 2014 (red line) with $\alpha = 0.02$ (0.02-0.02)
- Prop. of 2014 (red line) with $\alpha = 0.05$ (0.05-0.05)
- Prop. of 2014 (red line) with $\alpha = 0.1$ (0.1-0.1)
- Prop. of 2014 (red line) with $\alpha = 0.2$ (0.2-0.2)
- Prop. of 2014 (red line) with $\alpha = 0.5$ (0.5-0.5)
- Prop. of 2014 (red line) with $\alpha = 1.0$ (1.0-1.0)
- Prop. of 2014 (red line) with $\alpha = 2.0$ (2.0-2.0)
- Prop. of 2014 (red line) with $\alpha = 5.0$ (5.0-5.0)
- Prop. of 2014 (red line) with $\alpha = 10.0$ (10.0-10.0)
- Prop. of 2014 (red line) with $\alpha = 20.0$ (20.0-20.0)
- Prop. of 2014 (red line) with $\alpha = 50.0$ (50.0-50.0)
- Prop. of 2014 (red line) with $\alpha = 100.0$ (100.0-100.0)
- Prop. of 2014 (red line) with $\alpha = 200.0$ (200.0-200.0)
- Prop. of 2014 (red line) with $\alpha = 500.0$ (500.0-500.0)
- Prop. of 2014 (red line) with $\alpha = 1000.0$ (1000.0-1000.0)
- Prop. of 2014 (red line) with $\alpha = 2000.0$ (2000.0-2000.0)
- Prop. of 2014 (red line) with $\alpha = 5000.0$ (5000.0-5000.0)
- Prop. of 2014 (red line) with $\alpha = 10000.0$ (10000.0-10000.0)
- Prop. of 2014 (red line) with $\alpha = 20000.0$ (20000.0-20000.0)
- Prop. of 2014 (red line) with $\alpha = 50000.0$ (50000.0-50000.0)
- Prop. of 2014 (red line) with $\alpha = 100000.0$ (100000.0-100000.0)
- Prop. of 2014 (red line) with $\alpha = 200000.0$ (200000.0-200000.0)
- Prop. of 2014 (red line) with $\alpha = 500000.0$ (500000.0-500000.0)
- Prop. of 2014 (red line) with $\alpha = 1000000.0$ (1000000.0-1000000.0)
- Prop. of 2014 (red line) with $\alpha = 2000000.0$ (2000000.0-2000000.0)
- Prop. of 2014 (red line) with $\alpha = 5000000.0$ (5000000.0-5000000.0)
- Prop. of 2014 (red line) with $\alpha = 10000000.0$ (10000000.0-10000000.0)
- Prop. of 2014 (red line) with $\alpha = 20000000.0$ (20000000.0-20000000.0)
- Prop. of 2014 (red line) with $\alpha = 50000000.0$ (50000000.0-50000000.0)
- Prop. of 2014 (red line) with $\alpha = 100000000.0$ (100000000.0-100000000.0)
- Prop. of 2014 (red line) with $\alpha = 200000000.0$ (200000000.0-200000000.0)
- Prop. of 2014 (red line) with $\alpha = 500000000.0$ (500000000.0-500000000.0)
- Prop. of 2014 (red line) with $\alpha = 1000000000.0$ (1000000000.0-1000000000.0)
- Prop. of 2014 (red line) with $\alpha = 2000000000.0$ (2000000000.0-2000000000.0)
- Prop. of 2014 (red line) with $\alpha = 5000000000.0$ (5000000000.0-5000000000.0)
- Prop. of 2014 (red line) with $\alpha = 10000000000.0$ (10000000000.0-10000000000.0)
- Prop. of 2014 (red line) with $\alpha = 20000000000.0$ (20000000000.0-20000000000.0)
- Prop. of 2014 (red line) with $\alpha = 50000000000.0$ (50000000000.0-50000000000.0)
- Prop. of 2014 (red line) with $\alpha = 100000000000.0$ (100000000000.0-100000000000.0)
- Prop. of 2014 (red line) with $\alpha = 200000000000.0$ (200000000000.0-200000000000.0)
- Prop. of 2014 (red line) with $\alpha = 500000000000.0$ (500000000000.0-500000000000.0)
- Prop. of 2014 (red line) with $\alpha = 1000000000000.0$ (1000000000000.0-1000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 2000000000000.0$ (2000000000000.0-2000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 5000000000000.0$ (5000000000000.0-5000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 10000000000000.0$ (10000000000000.0-10000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 20000000000000.0$ (20000000000000.0-20000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 50000000000000.0$ (50000000000000.0-50000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 100000000000000.0$ (100000000000000.0-100000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 200000000000000.0$ (200000000000000.0-200000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 500000000000000.0$ (500000000000000.0-500000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 1000000000000000.0$ (1000000000000000.0-1000000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 2000000000000000.0$ (2000000000000000.0-2000000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 5000000000000000.0$ (5000000000000000.0-5000000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 10000000000000000.0$ (10000000000000000.0-10000000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 20000000000000000.0$ (20000000000000000.0-20000000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 50000000000000000.0$ (50000000000000000.0-50000000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 100000000000000000.0$ (100000000000000000.0-100000000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 200000000000000000.0$ (200000000000000000.0-200000000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 500000000000000000.0$ (500000000000000000.0-500000000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 1000000000000000000.0$ (1000000000000000000.0-1000000000000000000.0)
- Prop. of 2014 (red line) with $\alpha = 2000000000000000000.0$ (20

Figure 10 consists of three plots showing the effect of input CNN's mean probability (ξ) on the output of the proposed model. The x-axis for all plots is 'input CNN's mean probability (ξ)' ranging from 0.00 to 0.20. The y-axis for all plots is 'Output of the proposed model' on a log scale.

- Left Plot:** Shows the output of the proposed model for various models. The y-axis ranges from 10^0 to 10^2 . The models are:
 - Prop. E
 - Mean of 200 E of $\xi = 0.00$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.05$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.10$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.15$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.20$ ($\text{Prop. E} = 0.0000000000000000$)
- Middle Plot:** Shows the output of the proposed model for various models. The y-axis ranges from 10^0 to 10^2 . The models are:
 - Prop. E
 - Mean of 200 E of $\xi = 0.00$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.05$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.10$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.15$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.20$ ($\text{Prop. E} = 0.0000000000000000$)
- Right Plot:** Shows the output of the proposed model for various models. The y-axis ranges from 10^0 to 10^2 . The models are:
 - Prop. E
 - Mean of 200 E of $\xi = 0.00$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.05$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.10$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.15$ ($\text{Prop. E} = 0.0000000000000000$)
 - Mean of 200 E of $\xi = 0.20$ ($\text{Prop. E} = 0.0000000000000000$)

Figure 1 consists of two line plots comparing the proposed method (red line) with several existing methods (blue, green, purple, orange lines) across different input correlation mean probabilities (x).

Left Plot: Oxidized Weight (mg) vs Input Cor's mean's probability (x)

The y-axis represents Oxidized Weight (mg) from 0 to 700. The x-axis represents Input Cor's mean's probability (x) from 0 to 0.5. The legend includes:

- Prop. (red line)
- 2018 Liu, et al. (blue line)
- 2018 Liu, et al. (green line)
- 2018 Liu, et al. (purple line)
- 2018 Liu, et al. (orange line)
- 2018 Liu, et al. (blue line)
- 2018 Liu, et al. (green line)
- 2018 Liu, et al. (purple line)
- 2018 Liu, et al. (orange line)

Right Plot: Cor's Rate vs Input Cor's mean's probability (x)

The y-axis represents Cor's Rate from 0 to 200. The x-axis represents Input Cor's mean's probability (x) from 0 to 0.5. The legend includes:

- Prop. (red line)
- 2018 Liu, et al. (blue line)
- 2018 Liu, et al. (green line)
- 2018 Liu, et al. (purple line)
- 2018 Liu, et al. (orange line)
- 2018 Liu, et al. (blue line)
- 2018 Liu, et al. (green line)
- 2018 Liu, et al. (purple line)
- 2018 Liu, et al. (orange line)

Figure 1 consists of three subplots. Subplot (a) is a scatter plot titled 'Scatter Plot (Iteration in Legend)' showing 'Output Color's Probability (%)' on the y-axis (0.0 to 1.0) versus 'Input Color's Probability (%)' on the x-axis (0.0 to 1.0). It contains 1000 data series, each representing an iteration, showing a strong positive linear correlation. Subplot (b) is a line plot titled 'Avg. BIC (Iteration)' showing 'BIC' on the y-axis (67.0 to 81.0) versus 'Probability' on the x-axis (0.0 to 1.0). It displays 1000 data series for different iterations, showing a noisy trend that generally increases with probability. Subplot (c) is a line plot titled 'Color's Probability' showing 'Color's Probability' on the y-axis (0.0 to 1.0) versus 'Input Color's Probability (%)' on the x-axis (0.0 to 1.0). It compares the proposed model (blue line) with other models (various colored lines). The proposed model shows a sharp increase in probability as input probability increases, while other models remain relatively flat or show a much less pronounced increase.

Figure 10 displays the results of the proposed model for the 2014-2015 season, comparing the model's performance against the baseline (Random Guess) and the proposed model (Proposed Model) across various input case weakness probabilities (x).

The figure is divided into two main sections, each containing three subplots:

- Top Section (2014-2015 Season):**
 - Left Plot (ROC Curve):** Shows the True Positive Rate (y) versus the False Positive Rate (x) for the 2014-2015 season. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot includes a diagonal line representing the baseline (Random Guess) and a solid line representing the proposed model's performance. The proposed model's performance is significantly better than the baseline, indicating high predictive accuracy.
 - Middle Plot (ROC Curve):** Shows the True Positive Rate (y) versus the False Positive Rate (x) for the 2014-2015 season. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot includes a diagonal line representing the baseline (Random Guess) and a solid line representing the proposed model's performance. The proposed model's performance is significantly better than the baseline, indicating high predictive accuracy.
 - Right Plot (ROC Curve):** Shows the True Positive Rate (y) versus the False Positive Rate (x) for the 2014-2015 season. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot includes a diagonal line representing the baseline (Random Guess) and a solid line representing the proposed model's performance. The proposed model's performance is significantly better than the baseline, indicating high predictive accuracy.
- Bottom Section (2014-2015 Season):**
 - Left Plot (ROC Curve):** Shows the True Positive Rate (y) versus the False Positive Rate (x) for the 2014-2015 season. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot includes a diagonal line representing the baseline (Random Guess) and a solid line representing the proposed model's performance. The proposed model's performance is significantly better than the baseline, indicating high predictive accuracy.
 - Middle Plot (ROC Curve):** Shows the True Positive Rate (y) versus the False Positive Rate (x) for the 2014-2015 season. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot includes a diagonal line representing the baseline (Random Guess) and a solid line representing the proposed model's performance. The proposed model's performance is significantly better than the baseline, indicating high predictive accuracy.
 - Right Plot (ROC Curve):** Shows the True Positive Rate (y) versus the False Positive Rate (x) for the 2014-2015 season. The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The plot includes a diagonal line representing the baseline (Random Guess) and a solid line representing the proposed model's performance. The proposed model's performance is significantly better than the baseline, indicating high predictive accuracy.

The plots demonstrate that the proposed model consistently outperforms the baseline across all input case weakness probabilities, indicating its effectiveness in predicting the outcome of the 2014-2015 season.

Figure 1 displays six plots showing the performance of the proposed method for estimating the probability of a COVID-19 case being a contact. The plots are arranged in a 2x3 grid, comparing results for the year 2016 (top row) and 2014 (bottom row).

The left column shows the True Contact Rate (Y-axis) versus Input Case's Infection Probability (X-axis). The plots show a strong positive correlation between the input probability and the true contact rate, with the proposed method (red line) closely following the true contact rate across the range of input probabilities.

The middle column shows the Avg. Bias Computed (Y-axis) versus Probability (X-axis). The plots show that the average bias computed is generally low, indicating that the proposed method is unbiased across the range of input probabilities.

The right column shows the Conf. Rate (Y-axis) versus Input Case's Infection Probability (X-axis). The plots show that the confidence rate is generally high, indicating that the proposed method is confident in its estimates across the range of input probabilities.

The plots compare the performance of the proposed method (red line) against several other methods (blue, green, yellow, orange, and purple lines). The proposed method consistently shows the highest performance across all metrics and input probabilities.

Figure 10 displays the performance of the proposed model across different datasets and metrics. The figure is organized into a 3x3 grid of plots. The top row shows the 'Input Car's lease Probability (%)' on the x-axis versus the 'Output Car's lease Probability (%)' on the y-axis. The middle row shows the 'Avg. Big Consumed' on the x-axis versus the 'Probability' on the y-axis. The bottom row shows the 'Car's lease Number' on the x-axis versus the 'Input Car's lease Probability (%)' on the y-axis. Each plot includes a legend for various car models and years, such as '2019, 1, 200000', '2018, 1, 200000', etc.

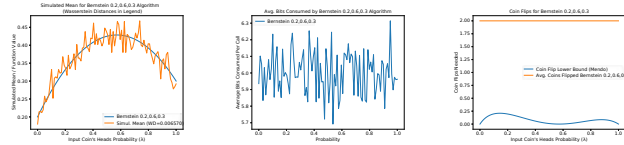
Figure 1 displays a 3x3 grid of plots showing the performance of the proposed method across different datasets and metrics. The rows represent datasets: 2019_3_00000, 2019_3_00000, and 2019_3_00000. The columns represent metrics: Squared Error, Avg. Bits Generated, and Cost Bits Generated. Each plot shows the relationship between input and output variables, with a legend indicating the specific model and its performance metrics.

[illegible]

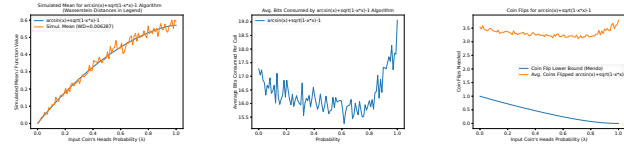
Figure 1 consists of three subplots, each showing the relationship between the Input Car's Health Probability (p) on the x-axis and the Output Car's Health Probability (q) on the y-axis. The x-axis for all plots ranges from 0.00 to 0.80. The y-axis for all plots ranges from 0.0 to 0.8.

- Top Plot:** The y-axis is labeled from 0.0 to 0.8. The curve starts at (0,0) and increases monotonically, passing through approximately (0.2, 0.2) and (0.4, 0.4), ending near (0.8, 0.7).
- Middle Plot:** The y-axis is labeled from 0.0 to 0.8. The curve starts at (0,0) and decreases monotonically, passing through approximately (0.2, 0.2) and (0.4, 0.4), ending near (0.8, 0.7).
- Bottom Plot:** The y-axis is labeled from 0.0 to 0.8. The curve starts at (0,0) and increases monotonically, passing through approximately (0.2, 0.2) and (0.4, 0.4), ending near (0.8, 0.7).

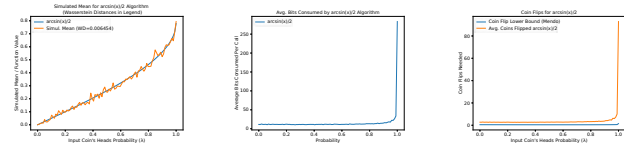
Bernstein
0.2,0.6,0.3



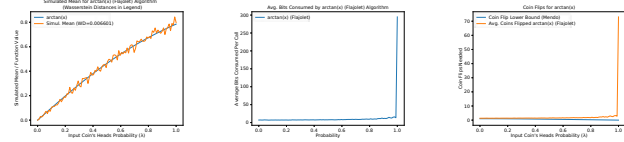
$\arcsin(x) + \sqrt{1-x^2} - 1$



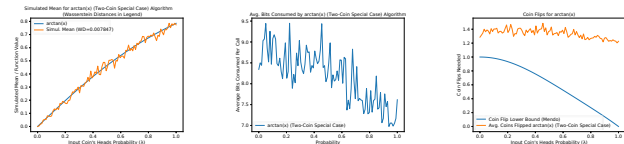
$\arcsin(x)/2$



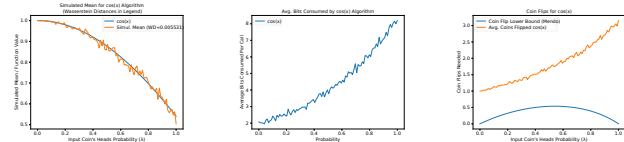
$\arctan(x)$
(Flajolet)



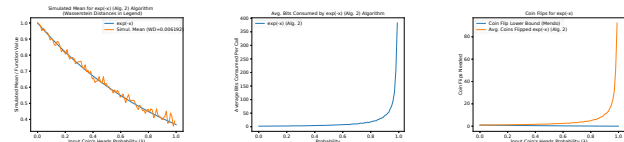
$\arctan(x)$ (Two-Coin Special Case)



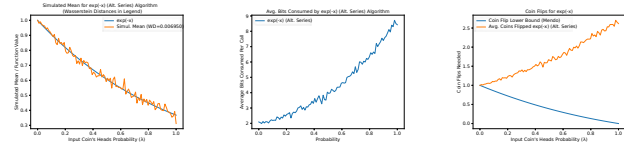
$\cos(x)$



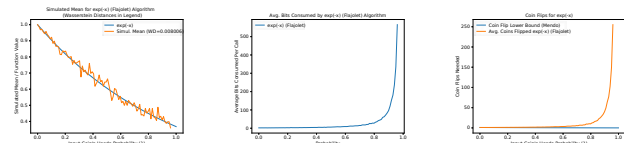
$\exp(-x)$ (Alg. 2)



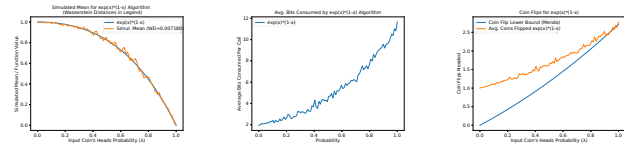
$\exp(-x)$ (Alt. Series)



$\exp(-x)$ (Flajolet)



$\exp(x) * (1-x)$



[illegible]

Figure 1 consists of three subplots labeled (a), (b), and (c), each showing performance metrics versus normalized frequency ω (ranging from 0.0 to 1.0).

- (a) Normalized Error:** The y-axis is 'Normalized Error (C/N) (dB)' from -10 to 0. The legend indicates 'Proposed' (blue line with circles) and 'Baseline' (orange line with circles). The proposed method shows a lower error than the baseline across the frequency range.
- (b) Average Number of Iterations:** The y-axis is 'Average Number of Iterations' from 0 to 10. The legend indicates 'Proposed' (blue line with circles) and 'Baseline' (orange line with circles). The proposed method requires fewer iterations than the baseline.
- (c) Coefficient Number:** The y-axis is 'Coeff. Number' from 0 to 1.5. The legend indicates 'Proposed' (blue line with circles) and 'Baseline' (orange line with circles). The proposed method has a lower coefficient number than the baseline.

Figure 1 consists of three subplots labeled (a), (b), and (c), each showing performance metrics for different algorithms across input correlations from 0.0 to 1.0.

- Subplot (a): Standard Error (SE) vs. Input Correlation**
 The y-axis is 'Standard Error (SE) x 10³' ranging from 0 to 1.5. The x-axis is 'Input Correlation (Input Correlation)' ranging from 0.0 to 1.0. The legend includes:
 - Estimated Mean for points 1/3/5 algorithm (black line with dots)
 - Importance Sampling (red line with dots)
 - Black Monte Carlo (blue line with dots)
 - point 1/3/5 (orange line with dots)
 The 'point 1/3/5' algorithm shows the highest SE, increasing from ~0.5 to ~1.4. The other algorithms show much lower SE, remaining below 0.2.
- Subplot (b): Avg. Bias Consumed vs. Input Correlation**
 The y-axis is 'Average Bias Consumed' ranging from 0 to 1600. The x-axis is 'Input Correlation' ranging from 0.0 to 1.0. The legend includes:
 - point 1/3/5 (orange line with dots)
 The 'point 1/3/5' algorithm shows a sharp drop in bias from ~1500 at correlation 0.0 to near 0 at correlation 0.2, remaining low thereafter.
- Subplot (c): CPU Time vs. Input Correlation**
 The y-axis is 'CPU Time (seconds)' ranging from 0 to 400. The x-axis is 'Input Correlation (Input Correlation)' ranging from 0.0 to 1.0. The legend includes:
 - Coin Flip for points 1/3/5 (blue line with dots)
 - Coin Flip for case-based (black line with dots)
 - Coin Flip for input points 1/3/5 (red line with dots)
 All three algorithms show a sharp drop in CPU time from ~350 seconds at correlation 0.0 to near 0 at correlation 0.2, remaining low thereafter.

Figure 10 consists of three subplots comparing the proposed algorithm (blue line) with the baseline algorithm (orange line) across three datasets: *gm1235*, *gm1236*, and *gm1237*. The x-axis for all plots is 'Input Correlation Coefficient (ρ)' ranging from 0.0 to 1.0. The y-axis for the left plot is 'Decrease in the Number of Iterations' (0 to 1.5), for the middle plot is 'Avg. Bits Consumed by prop. 2/3E algorithm' (0 to 300), and for the right plot is 'CPU Time (s)' (0 to 100). In all cases, the proposed algorithm shows a significant improvement over the baseline algorithm.

Figure 1 consists of three subplots labeled (a), (b), and (c).

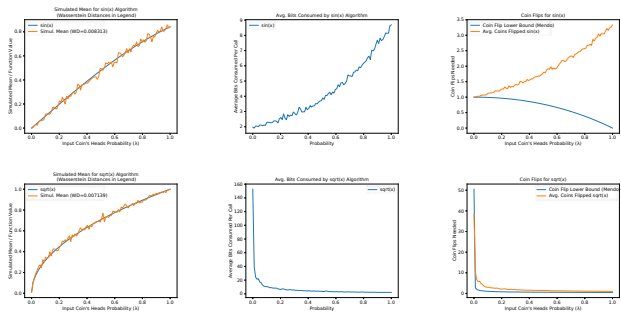
- (a) ROC curve for average AUC against input Cori's records probability:** The x-axis is 'input Cori's records probability (0.0-1.0)' and the y-axis is 'Data point's True rate (0.0-1.0)'. A red line represents the 'Actual, Mean (0.9141-0.998863)' and a blue line represents the 'Ideal Mean for average AUC against input Cori's records probability'. The red line is above the blue line, indicating better performance.
- (b) Mean time consumed by parallel algorithm against probability:** The x-axis is 'Probability' and the y-axis is 'Mean Time Consumed (Sec) (0.0-45)'. A blue line represents 'Parallel, 100' and a red line represents 'Parallel, 1000'. Both lines show a sharp decrease in time as probability increases, with the 1000 parallel lines performing faster.
- (c) Cori flag for parallel algorithm against probability:** The x-axis is 'input Cori's records probability (0.0-1.0)' and the y-axis is 'Cori flag (0.0-1.0)'. A blue line represents 'Cori flag for parallel, 100' and a red line represents 'Cori flag for parallel, 1000'. Both lines show a sharp decrease in the flag value as probability increases, with the 1000 parallel lines performing better.

[illegible]

Figure 1 consists of four plots arranged in a 2x2 grid, showing the performance of different methods. The top-left plot shows the Discretized Area Under the Curve (AUC) for the 'area' variable against the Input Data's Feature Probability (x). The top-right plot shows the Average Area Under the Curve (AUC) for the 'area' variable against the Probability. The bottom-left plot shows the Average Area Under the Curve (AUC) for the 'area' variable against the Probability. The bottom-right plot shows the CDF of the Error for the 'area' variable against the Input Data's Feature Probability (x).

Figure 10 consists of three subplots. The left plot shows the Standard Error (Y-axis, 0.0 to 0.4) versus Frequency (Hz) (X-axis, 0.0 to 1.0). It compares the proposed model at 5.0 Hz (blue line with circles) and 5.5 Hz (orange line with circles). Both models show a linear increase in error with frequency, with the 5.5 Hz model having a slightly higher error. The middle plot shows the Magnitude of Transfer Function $G(s)$ (Y-axis, 0.0 to 1.0) versus Frequency (Hz) (X-axis, 0.0 to 1.0). The proposed model at 5.0 Hz (blue line) shows a sharp drop in magnitude from 1.0 to near 0.0 at a frequency of approximately 0.1 Hz. The right plot shows the Cost Function (Y-axis, 0.0 to 1200) versus Frequency (Hz) (X-axis, 0.0 to 1.0). It compares the proposed model at 5.0 Hz (blue line) with a lower bound (orange line) and a reference model (green line). The proposed model's cost function drops sharply from approximately 1100 to near 0.0 at a frequency of approximately 0.1 Hz, while the lower bound and reference model remain near 0.0.

$\sin(x)$



\sqrt{x}

1. <https://peteroupc.github.io/bernoulli.md>