

# 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**<sup>1</sup>” and show their performance in terms of the number of bits they use on average. For each algorithm, and for each of 100  $\lambda$  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  $\lambda$  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  $\lambda$ , the entire data point for that  $\lambda$  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)<sup>[^1]</sup>. Note that some functions require a growing number of coin flips as  $\lambda$  approaches 0 or 1. Note that for the 2014, 2016, and 2019 algorithms—

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

Points with invalid  $\epsilon$  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

---

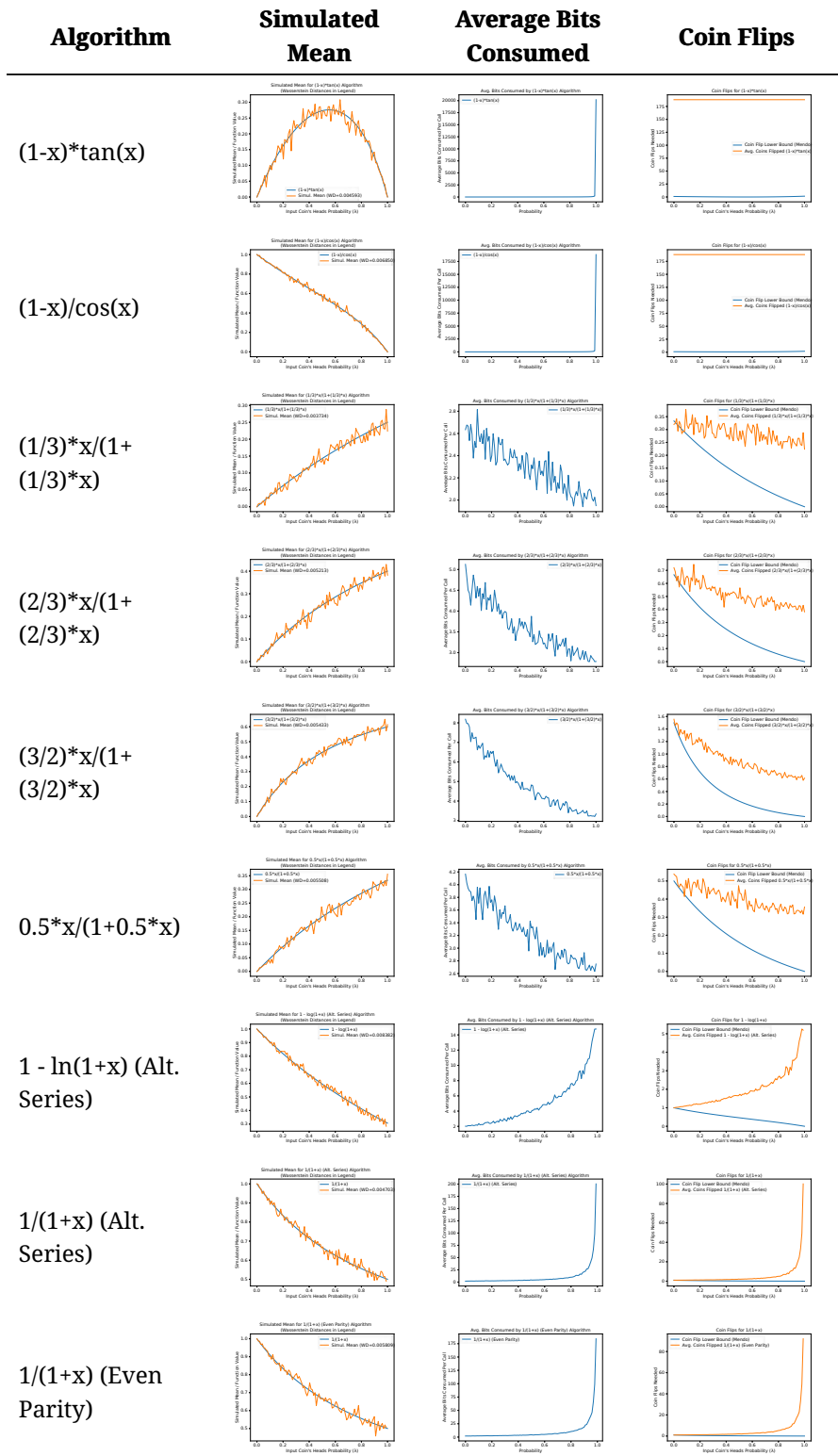


Figure 10 consists of four subplots arranged in a 2x2 grid, all sharing the same x-axis: 'Input data bandwidth probability (%)' ranging from 0.0 to 1.0.

- Top-left plot:** Titled 'Average Mean Squared Error (MSE)'. The y-axis ranges from 0.000 to 0.004. It shows a blue line with markers representing the 'Error' and a black line representing the 'Error Mean (MSE) = 0.00254'. The error fluctuates around the mean, which decreases slightly as bandwidth probability increases.
- Top-right plot:** Titled 'Avg. Bits Consumed by SLS vs. Input data bandwidth probability'. The y-axis ranges from 3.0 to 6.0. It shows a blue line with markers representing the 'Error' and a black line representing the 'Error Mean (SLS) = 5.019'. The error fluctuates around the mean, which decreases as bandwidth probability increases.
- Bottom-left plot:** Titled 'Error Rate (BER)'. The y-axis ranges from 0.000 to 0.004. It shows a blue line with markers representing the 'Error' and a black line representing the 'Error Mean (BER) = 0.00254'. The error fluctuates around the mean, which decreases slightly as bandwidth probability increases.
- Bottom-right plot:** Titled 'Error Rate (BER)'. The y-axis ranges from 0.00 to 0.05. It shows two lines: a black line for 'Lower Bound (Black)' and an orange line for 'Upper Bound (Orange)'. The black line is relatively flat and low, while the orange line fluctuates significantly between approximately 0.01 and 0.04.

Figure 1 consists of three subplots labeled (a), (b), and (c), all sharing the same x-axis representing the  $L5L\text{-}wr$  parameter range from 1.0 to 1.6.

- (a) Simulated Mean for  $L5L\text{-}wr$  algorithm:** This plot shows the Simulated Mean (orange line) and its 95% Confidence Interval (grey shaded area). The y-axis is labeled "Simulated Mean (95% CI)" and ranges from 0.0 to 0.4. The mean value starts around 0.25 at  $L5L\text{-}wr = 1.0$  and decreases to approximately 0.15 at  $L5L\text{-}wr = 1.6$ .
- (b) Avg. RMSE for  $L5L\text{-}wr$  algorithm:** This plot shows the Average Root Mean Square Error (RMSE) in blue. The y-axis is labeled "Avg. RMSE (95% CI)" and ranges from 0.0 to 0.6. The RMSE starts around 0.55 at  $L5L\text{-}wr = 1.0$  and decreases to approximately 0.25 at  $L5L\text{-}wr = 1.6$ .
- (c) Cor. Rsp. for  $L5L\text{-}wr$  algorithm:** This plot shows the Correlation Response (Cor. Rsp.) in blue. The y-axis is labeled "Cor. Rsp. (95% CI)" and ranges from 0.00 to 0.20. The correlation response is consistently low, staying near 0.00 across the entire range of  $L5L\text{-}wr$ .

Figure 1 consists of three subplots. The left subplot, titled 'Error Absolute Value (log scale)', shows the error for various datasets (e.g., 2014\_2\_200000, 2014\_2\_2000000) plotted against 'Avg. Bids Consumed'. The middle subplot, titled 'Average Difference between Cdf', shows the difference between the cumulative distribution functions for the same datasets. The right subplot, titled 'Cost Ratio', shows the 'Cost Ratio Upper Bound' and 'Cost Ratio Lower Bound' for the 2014\_1\_200000 dataset against 'Avg. Bids Consumed'.

[illegible]

Figure 1 consists of three subplots. Subplot (a) is a Receiver Operating Characteristic (ROC) curve for the 2014-2015 season, showing the True Positive Rate (Y-axis) versus the False Positive Rate (X-axis). The curve is a solid blue line, indicating good performance. Subplot (b) is a line graph showing the Average AUC over 1000 trials for the 2014-2015 season. The Y-axis is 'Average AUC over 1000 trials' (0.6 to 0.9) and the X-axis is 'Run' (0 to 1000). Multiple colored lines represent different models, with a legend on the right. Subplot (c) is a line graph showing the Cumulative Number of cases (Y-axis, 0 to 1000) versus 'Run' (X-axis, 0 to 1000). The legend on the right indicates that the blue line represents 'Cumulative number of cases for 2014-2015 season'.

[illegible]

Figure 1 consists of three subplots comparing the proposed method (2014, S. Gotoh et al.) with other state-of-the-art methods across different sample sizes (0 to 10,000).

**Left Subplot: Error Rate (Mean) - F1 on the Edge**

This plot shows the error rate (Mean) - F1 on the Edge versus the Number of Samples. The proposed method (2014, S. Gotoh et al.) consistently achieves the lowest error rate across all sample sizes, starting around 0.05 and decreasing to approximately 0.01 as the number of samples increases.

**Middle Subplot: Rank Size (Component)**

This plot shows the Rank Size (Component) versus the Number of Samples. The proposed method (2014, S. Gotoh et al.) maintains a low rank size, generally below 1000, while other methods show significantly higher and more volatile rank sizes, often exceeding 2000.

**Right Subplot: CPU Time (Second)**

This plot shows the CPU Time (Second) versus the Number of Samples. The proposed method (2014, S. Gotoh et al.) has a very low and stable CPU time, around 10 seconds, whereas other methods show a sharp increase in CPU time as the number of samples increases, reaching up to 1000 seconds for some methods.

Figure 1 consists of three subplots. The left subplot, titled 'Circumference Mean Squared Error (Proposed Model Circumference)', shows 'Normalized Mean Squared Error' on the y-axis (0.0 to 0.4) versus 'Normalized Feature Number' on the x-axis (0.0 to 1.0). It compares 'Proposed Model (Circumference)' (blue line) and 'Proposed Model (Area)' (green line). The middle subplot, titled 'Avg. Step Circumference', shows 'Avg. Step Circumference' on the y-axis (0 to 250) versus 'Step' on the x-axis (0 to 100). It compares '2016 Area v=0.2' (blue line), '2016 Area v=0.3' (orange line), and '2016 Area v=0.2' (green line). The right subplot, titled 'Circ. Step Lower Bound', shows 'Accuracy (Percent)' on the y-axis (0 to 100) versus 'Step' on the x-axis (0 to 100). It compares 'Circ. Step Lower Bound Circumference' (blue line), 'Circ. Step Lower Bound Area' (orange line), 'Avg. Step Circumference 2016 Area v=0.2' (green line), and 'Avg. Step Circumference 2016 Area v=0.3' (red line).

Figure 1 consists of three subplots comparing the proposed method with state-of-the-art methods. The left subplot shows the Spearman Rank of the proposed method versus the Spearman Rank of the state-of-the-art methods. The middle subplot shows the Average Significance Countdown of the proposed method versus the Average Significance Countdown of the state-of-the-art methods. The right subplot shows the Cost (Average Rate) of the proposed method versus the Cost (Average Rate) of the state-of-the-art methods. In all three plots, the proposed method (blue line) shows superior performance compared to the state-of-the-art methods (colored lines).

[illegible]

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, (a), (b), and (c), each showing the CAGR (Y-axis) versus the input CAGR's means Probability (X-axis). The X-axis for all plots ranges from 0.0 to 1.0. The Y-axis for (a) ranges from 0 to 400, for (b) from 0 to 1400, and for (c) from 0 to 1400. Each plot compares the proposed model (blue line) with other models (colored lines) across different input CAGR means and probabilities. The proposed model generally shows a more stable and accurate prediction compared to the other models, especially in the wet weather scenario.

**(a) Dry Weather (CAGR vs. Prob)**

Legend:

- Prop. (Blue line)
- 2014-16 (Green line)
- Mean of 2014-16,  $\mu=0.102$  ( $\sigma=0.0289$ ) (Red line)
- Mean of 2014-16,  $\mu=0.102$  ( $\sigma=0.0355$ ) (Orange line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.0$  (Purple line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.05$  (Yellow line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.1$  (Pink line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.15$  (Light Blue line)

**(b) Wet Weather (CAGR vs. Prob)**

Legend:

- Prop. (Blue line)
- 2014-16 (Green line)
- Mean of 2014-16,  $\mu=0.102$  ( $\sigma=0.0289$ ) (Red line)
- Mean of 2014-16,  $\mu=0.102$  ( $\sigma=0.0355$ ) (Orange line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.0$  (Purple line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.05$  (Yellow line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.1$  (Pink line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.15$  (Light Blue line)

**(c) CAGR (CAGR vs. Prob)**

Legend:

- Prop. (Blue line)
- 2014-16 (Green line)
- Mean of 2014-16,  $\mu=0.102$  ( $\sigma=0.0289$ ) (Red line)
- Mean of 2014-16,  $\mu=0.102$  ( $\sigma=0.0355$ ) (Orange line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.0$  (Purple line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.05$  (Yellow line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.1$  (Pink line)
- Low Mean,  $\mu=0.05$ ,  $\sigma=0.15$  (Light Blue line)

Figure 10 consists of three subplots showing the effect of input CNN's mean probability ( $\xi$ ) on the output of the proposed model. The x-axis for all plots is 'Input CNN's mean Probability ( $\xi$ )' ranging from 0.00 to 0.6. The y-axis for all plots is 'Output of the proposed model'.

- Left Plot:** The y-axis ranges from 0 to 1. It shows a single curve that increases from 0 to 1 as  $\xi$  increases. The legend indicates:
  - Prop. (black line)
  - Mean of 2014-15  $\mu=0.10$  (0.03852) (red line)
  - Mean of 2014-15  $\mu=0.10$  (0.03852) (blue line)
  - Mean of 2014-15  $\mu=0.10$  (0.03852) (green line)
  - Mean of 2014-15  $\mu=0.10$  (0.03852) (purple line)
- Middle Plot:** The y-axis ranges from 0 to 400. It shows multiple curves that increase as  $\xi$  increases. The legend indicates:
  - 2014-15  $\mu=2$  (black line)
  - 2014-15  $\mu=2$  (red line)
  - 2014-15  $\mu=2$  (blue line)
  - 2014-15  $\mu=2$  (green line)
  - 2014-15  $\mu=2$  (purple line)
  - Low Mean  $\mu=1$   $\mu=0.5$  (black line)
  - Low Mean  $\mu=1$   $\mu=0.5$  (red line)
  - Low Mean  $\mu=1$   $\mu=0.5$  (blue line)
  - Low Mean  $\mu=1$   $\mu=0.5$  (green line)
  - Low Mean  $\mu=1$   $\mu=0.5$  (purple line)
- Right Plot:** The y-axis ranges from 0 to 1400. It shows multiple curves that increase as  $\xi$  increases. The legend indicates:
  - Low-Top Low Mean  $\mu=0.5$  (black line)
  - Low-Top Low Mean  $\mu=0.5$  (red line)
  - Low-Top Low Mean  $\mu=0.5$  (blue line)
  - Low-Top Low Mean  $\mu=0.5$  (green line)
  - Low-Top Low Mean  $\mu=0.5$  (purple line)
  - Low-Top Low Mean  $\mu=0.5$  (black line)
  - Low-Top Low Mean  $\mu=0.5$  (red line)
  - Low-Top Low Mean  $\mu=0.5$  (blue line)
  - Low-Top Low Mean  $\mu=0.5$  (green line)
  - Low-Top Low Mean  $\mu=0.5$  (purple line)

Figure 1 consists of two line plots comparing the proposed method (red line) with several existing methods (blue, green, orange, purple, brown lines) across different input Cor's mean's probability (x) values (0.0, 0.1, 0.2, 0.3, 0.4, 0.5).

**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 0.0 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. (orange line)
- 2018 Liu, et al. (purple line)
- 2018 Liu, et al. (brown line)
- 2018 Liu, et al. (black line)
- 2018 Liu, et al. (grey line)
- 2018 Liu, et al. (dark blue line)
- 2018 Liu, et al. (dark green line)
- 2018 Liu, et al. (dark orange line)
- 2018 Liu, et al. (dark purple line)
- 2018 Liu, et al. (dark brown line)
- 2018 Liu, et al. (dark grey line)
- 2018 Liu, et al. (dark dark blue line)
- 2018 Liu, et al. (dark dark green line)
- 2018 Liu, et al. (dark dark orange line)
- 2018 Liu, et al. (dark dark purple line)
- 2018 Liu, et al. (dark dark brown line)
- 2018 Liu, et al. (dark dark grey line)

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

The y-axis represents Cor's Mean from 0 to 200. The x-axis represents Input Cor's mean's probability (x) from 0.0 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. (orange line)
- 2018 Liu, et al. (purple line)
- 2018 Liu, et al. (brown line)
- 2018 Liu, et al. (black line)
- 2018 Liu, et al. (grey line)
- 2018 Liu, et al. (dark blue line)
- 2018 Liu, et al. (dark green line)
- 2018 Liu, et al. (dark orange line)
- 2018 Liu, et al. (dark purple line)
- 2018 Liu, et al. (dark brown line)
- 2018 Liu, et al. (dark grey line)
- 2018 Liu, et al. (dark dark blue line)
- 2018 Liu, et al. (dark dark green line)
- 2018 Liu, et al. (dark dark orange line)
- 2018 Liu, et al. (dark dark purple line)
- 2018 Liu, et al. (dark dark brown line)
- 2018 Liu, et al. (dark dark grey line)

Figure 1 consists of three subplots. Subplot (a) is a scatter plot titled 'Scatter Plot Between Input and Output' showing a strong positive linear correlation between 'Input Color's Resale Probability (%)' and 'Output Color's Resale Probability (%)'. Subplot (b) is a line plot titled 'Accuracy vs. Probability' showing the accuracy of various models across different probability ranges. The models include Color-Flipped Resale Probability, Color-Flipped Resale Probability + 200000, Color-Flipped Resale Probability + 100000, Color-Flipped Resale Probability + 50000, Color-Flipped Resale Probability + 25000, Color-Flipped Resale Probability + 12500, Color-Flipped Resale Probability + 6250, Color-Flipped Resale Probability + 3125, Color-Flipped Resale Probability + 1562, Color-Flipped Resale Probability + 781, Color-Flipped Resale Probability + 390, Color-Flipped Resale Probability + 195, Color-Flipped Resale Probability + 97, Color-Flipped Resale Probability + 48, Color-Flipped Resale Probability + 24, Color-Flipped Resale Probability + 12, Color-Flipped Resale Probability + 6, Color-Flipped Resale Probability + 3, Color-Flipped Resale Probability + 1, Color-Flipped Resale Probability + 0.5, Color-Flipped Resale Probability + 0.25, Color-Flipped Resale Probability + 0.125, Color-Flipped Resale Probability + 0.0625, Color-Flipped Resale Probability + 0.03125, Color-Flipped Resale Probability + 0.015625, Color-Flipped Resale Probability + 0.0078125, Color-Flipped Resale Probability + 0.00390625, Color-Flipped Resale Probability + 0.001953125, Color-Flipped Resale Probability + 0.0009765625, Color-Flipped Resale Probability + 0.00048828125, Color-Flipped Resale Probability + 0.000244140625, Color-Flipped Resale Probability + 0.0001220703125, Color-Flipped Resale Probability + 0.00006103515625, Color-Flipped Resale Probability + 0.000030517578125, Color-Flipped Resale Probability + 0.0000152587890625, Color-Flipped Resale Probability + 0.00000762939453125, Color-Flipped Resale Probability + 0.000003814697265625, Color-Flipped Resale Probability + 0.0000019073486328125, Color-Flipped Resale Probability + 0.00000095367431640625, Color-Flipped Resale Probability + 0.000000476837158203125, Color-Flipped Resale Probability + 0.0000002384185791015625, Color-Flipped Resale Probability + 0.00000011920928955078125, Color-Flipped Resale Probability + 0.000000059604644775390625, Color-Flipped Resale Probability + 0.0000000298023223876953125, Color-Flipped Resale Probability + 0.00000001490116119384765625, Color-Flipped Resale Probability + 0.000000007450580596923828125, Color-Flipped Resale Probability + 0.0000000037252902984619140625, Color-Flipped Resale Probability + 0.00000000186264514923095703125, Color-Flipped Resale Probability + 0.000000000931322574615478515625, Color-Flipped Resale Probability + 0.0000000004656612873077392578125, Color-Flipped Resale Probability + 0.00000000023283064365386962890625, Color-Flipped Resale Probability + 0.000000000116415321826934814453125, Color-Flipped Resale Probability + 0.0000000000582076609134674072265625, Color-Flipped Resale Probability + 0.00000000002910383045673370361328125, Color-Flipped Resale Probability + 0.000000000014551915228366851806640625, Color-Flipped Resale Probability + 0.0000000000072759576141834259033203125, Color-Flipped Resale Probability + 0.00000000000363797880709171295166015625, Color-Flipped Resale Probability + 0.000000000001818989403545856475830078125, Color-Flipped Resale Probability + 0.0000000000009094947017729282379150390625, Color-Flipped Resale Probability + 0.00000000000045474735088646411895751953125, Color-Flipped Resale Probability + 0.000000000000227373675443232059478759765625, Color-Flipped Resale Probability + 0.0000000000001136868377216160297393798828125, Color-Flipped Resale Probability + 0.00000000000005684341886080801486968994140625, Color-Flipped Resale Probability + 0.000000000000028421709430404007434844970703125, Color-Flipped Resale Probability + 0.0000000000000142108547152020037174224853515625, Color-Flipped Resale Probability + 0.00000000000000710542735760100185871124267578125, Color-Flipped Resale Probability + 0.000000000000003552713678800500929355621337890625, Color-Flipped Resale Probability + 0.0000000000000017763568394002504646778106689453125, Color-Flipped Resale Probability + 0.00000000000000088817841970012523233890533447265625, Color-Flipped Resale Probability + 0.000000000000000444089209850062616169452667236328125, Color-Flipped Resale Probability + 0.0000000000000002220446049250313080847263336181640625, Color-Flipped Resale Probability + 0.00000000000000011102230246251565404236316680908203125, Color-Flipped Resale Probability + 0.000000000000000055511151231257827021181583340541015625, Color-Flipped Resale Probability + 0.0000000000000000277555756156289135105907916702705078125, Color-Flipped Resale Probability + 0.00000000000000001387778780781445675529539583513525390625, Color-Flipped Resale Probability + 0.000000000000000006938893903907228377647697917567626953125, Color-Flipped Resale Probability + 0.0000000000000000034694469519536141888238489587838134765625, Color-Flipped Resale Probability + 0.00000000000000000173472347597680709441192447939190673828125, Color-Flipped Resale Probability + 0.000000000000000000867361737988403547205962239695953369140625, Color-Flipped Resale Probability + 0.0000000000000000004336808689942017736029811198479766845703125, Color-Flipped Resale Probability + 0.00000000000000000021684043449710088680149055992398834228515625, Color-Flipped Resale Probability + 0.000000000000000000108420217248550443400745279961994171142578125, Color-Flipped Resale Probability + 0.0000000000000000000542101086242752217003726399809970855712890625, Color-Flipped Resale Probability + 0.00000000000000000002710505431213761085018631999049854278564453125, Color-Flipped Resale Probability + 0.000000000000000000013552527156068805425093159995249271392822265625, Color-Flipped Resale Probability + 0.000000000000000000006776263578034402712546579997624635696411328125, Color-Flipped Resale Probability + 0.0000000000000000000033881317890172013562732899988123178482056640625, Color-Flipped Resale Probability + 0.0000000000000000000016940658945086006781366449994061592241028203125, Color-Flipped Resale Probability + 0.00000000000000000000084703294725430033906832249970307961205141015625, Color-Flipped Resale Probability + 0.00000000000000000000042351647362715016953416124985153980602570578125, Color-Flipped Resale Probability + 0.000000000000000000000211758236813575084767080624925769903012852890625, Color-Flipped Resale Probability + 0.0000000000000000000001058791184067875423835403124628849515064264453125, Color-Flipped Resale Probability + 0.00000000000000000000005293955920339377119177015623114247575321322265625, Color-Flipped Resale Probability + 0.00000000000000000000002646977960169688559588507811557

Figure 10 consists of six subplots arranged in a 3x2 grid, showing the performance of the proposed model on the 2014 and 2015 datasets. The left column displays 'Spilloverd Means' (Normalized Distance) and the right column displays 'Avg. Bins Consumed' (Avg. Bins Consumed Per CoV) for both datasets. Each plot includes a legend with 10 entries representing different spilloverd means and their corresponding input CoV's weak probability ranges.

**Top Row (2014 Dataset):**

- Spilloverd Means (Normalized Distance vs. Input CoV's Weak Probability):** The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The legend lists 10 spilloverd means with their corresponding input CoV's weak probability ranges.
- Avg. Bins Consumed (Avg. Bins Consumed Per CoV vs. Probability):** The y-axis ranges from 0.0 to 100.0, and the x-axis ranges from 0.0 to 0.5. The legend lists 10 spilloverd means with their corresponding input CoV's weak probability ranges.

**Bottom Row (2015 Dataset):**

- Spilloverd Means (Normalized Distance vs. Input CoV's Weak Probability):** The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 0.5. The legend lists 10 spilloverd means with their corresponding input CoV's weak probability ranges.
- Avg. Bins Consumed (Avg. Bins Consumed Per CoV vs. Probability):** The y-axis ranges from 0.0 to 100.0, and the x-axis ranges from 0.0 to 0.5. The legend lists 10 spilloverd means with their corresponding input CoV's weak probability ranges.

The plots show that the proposed model's performance is consistent across different datasets and spilloverd means. The 'Spilloverd Means' plots show a strong positive correlation between the normalized distance and the input CoV's weak probability. The 'Avg. Bins Consumed' plots show that the average number of bins consumed per CoV is relatively low and stable across different input CoV's weak probabilities.

Figure 1 consists of six subplots arranged in a 2x3 grid, showing the performance of the proposed model across different years (2014-2020) and various input parameters.

- Top Left:** Scatter plot titled "Observed Mean (Weaverburn Database is Legend)" showing the relationship between "Input Class Imbalance Probability (%)". The y-axis is "Observed Mean" (0.00 to 1.00) and the x-axis is "Input Class Imbalance Probability (%)" (0.00 to 1.00). The plot shows a strong positive correlation between the observed mean and the input class imbalance probability.
- Top Middle:** Line plot titled "Avg. Size Consumed" showing the relationship between "Probability" (0.00 to 1.00) and "Average F1 Score Consumed (%)". The y-axis ranges from 160 to 190. The plot shows multiple lines for different years (2014-2020) and epochs (10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000, 90000, 100000). The F1 score generally increases with probability, with some fluctuations.
- Top Right:** Line plot titled "Cost Ratio for 2014, 100000" showing the relationship between "Input Class Imbalance Probability (%)" (0.00 to 1.00) and "Cost Ratio Consumed" (0 to 100). The y-axis ranges from 0 to 100. The plot shows multiple lines for different years (2014-2020) and epochs (10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000, 90000, 100000). The cost ratio generally increases with input class imbalance probability, with some fluctuations.
- Bottom Left:** Scatter plot titled "Observed Mean (Weaverburn Database is Legend)" showing the relationship between "Input Class Imbalance Probability (%)". The y-axis is "Observed Mean" (0.00 to 1.00) and the x-axis is "Input Class Imbalance Probability (%)" (0.00 to 1.00). The plot shows a strong positive correlation between the observed mean and the input class imbalance probability.
- Bottom Middle:** Line plot titled "Avg. Size Consumed" showing the relationship between "Probability" (0.00 to 1.00) and "Average F1 Score Consumed (%)". The y-axis ranges from 160 to 190. The plot shows multiple lines for different years (2014-2020) and epochs (10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000, 90000, 100000). The F1 score generally increases with probability, with some fluctuations.
- Bottom Right:** Line plot titled "Cost Ratio for 2014, 100000" showing the relationship between "Input Class Imbalance Probability (%)" (0.00 to 1.00) and "Cost Ratio Consumed" (0 to 100). The y-axis ranges from 0 to 100. The plot shows multiple lines for different years (2014-2020) and epochs (10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000, 90000, 100000). The cost ratio generally increases with input class imbalance probability, with some fluctuations.

Figure 1 displays nine plots arranged in a 3x3 grid, showing the relationship between input and output variables for different models and datasets. The rows represent different models: 'Simplified Model', 'Aug. 8th Consumed', and 'Call Risk for 2015, 1,00000'. The columns represent different datasets: 'Simplified Model', 'Aug. 8th Consumed', and 'Call Risk for 2015, 1,00000'. Each plot shows a scatter of data points and a fitted curve. The x-axis for all plots is 'Input Call's Net Profit Probability (%)' and the y-axis is 'Output Call's Net Profit Probability (%)'. The plots show that the model's output is generally higher than the input, indicating a positive bias.

Figure 10 displays the comparison of the proposed model with the baseline model across three datasets: 2019-2, 2019-3, and 2019-4. The figure is organized into three rows, each corresponding to a dataset, and two columns, each showing a different metric.

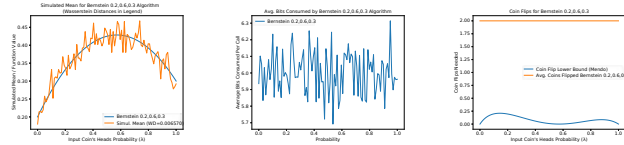
The left column shows the **Calibration** plots, which compare the **True Mean** (Y-axis) against the **Input Data Mean Probability (x)** (X-axis). The plots show the performance of the **Baseline Model** (grey line) and the **Proposed Model** (blue line) across various input data sets (e.g., 2019-1, 2019-2, 2019-3, 2019-4, 2019-5, 2019-6, 2019-7, 2019-8, 2019-9, 2019-10, 2019-11, 2019-12, 2019-13, 2019-14, 2019-15, 2019-16, 2019-17, 2019-18, 2019-19, 2019-20, 2019-21, 2019-22, 2019-23, 2019-24, 2019-25, 2019-26, 2019-27, 2019-28, 2019-29, 2019-30, 2019-31, 2019-32, 2019-33, 2019-34, 2019-35, 2019-36, 2019-37, 2019-38, 2019-39, 2019-40, 2019-41, 2019-42, 2019-43, 2019-44, 2019-45, 2019-46, 2019-47, 2019-48, 2019-49, 2019-50, 2019-51, 2019-52, 2019-53, 2019-54, 2019-55, 2019-56, 2019-57, 2019-58, 2019-59, 2019-60, 2019-61, 2019-62, 2019-63, 2019-64, 2019-65, 2019-66, 2019-67, 2019-68, 2019-69, 2019-70, 2019-71, 2019-72, 2019-73, 2019-74, 2019-75, 2019-76, 2019-77, 2019-78, 2019-79, 2019-80, 2019-81, 2019-82, 2019-83, 2019-84, 2019-85, 2019-86, 2019-87, 2019-88, 2019-89, 2019-90, 2019-91, 2019-92, 2019-93, 2019-94, 2019-95, 2019-96, 2019-97, 2019-98, 2019-99, 2019-100). The proposed model (blue line) shows significantly better calibration than the baseline model (grey line).

The right column shows the **Calibration Error** plots, which compare the **Calibration Error** (Y-axis) against the **Input Data Mean Probability (x)** (X-axis). The plots show the performance of the **Baseline Model** (grey line) and the **Proposed Model** (blue line) across various input data sets. The proposed model (blue line) shows significantly lower calibration error than the baseline model (grey line).

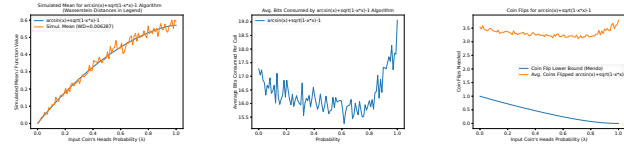
Figure 1 displays six plots showing the relationship between input and output variables for the 2019 COVID-19 dataset. The plots are arranged in a 3x2 grid. The top row shows 'Avg. All Compartment' vs 'Input Case Incidence Probability (%)'. The middle row shows 'Avg. All Compartment' vs 'Input Case Incidence Probability (%)'. The bottom row shows 'Avg. All Compartment' vs 'Input Case Incidence Probability (%)'. Each plot includes a legend for 'Generated Values' and 'Observed Values'.

Figure 1 consists of three subplots. The top-left plot shows the output GCM's network probability (y-axis, 0.0 to 0.2) versus the input GCM's network probability (x-axis, 0.00 to 0.90). It contains several colored lines (green, blue, red, orange, purple) that all start at (0,0) and increase monotonically, with some lines being steeper than others. The top-right plot shows the average path length (y-axis, 100 to 150) versus probability (x-axis, 0.00 to 0.90). It contains several colored lines that generally decrease as probability increases, with some lines showing a more pronounced dip around 0.5 probability. The bottom plot shows the average clustering coefficient (y-axis, 0 to 20) versus the input GCM's network probability (x-axis, 0.00 to 0.90). It contains several colored lines that generally increase as probability increases, with some lines showing a more pronounced increase around 0.5 probability.

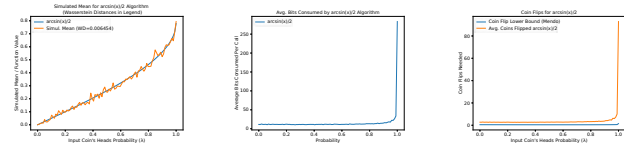
Bernstein  
0.2,0.6,0.3



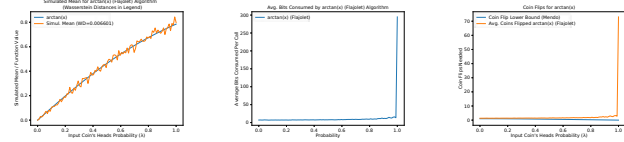
$\arcsin(x) + \sqrt{1-x^2} - 1$



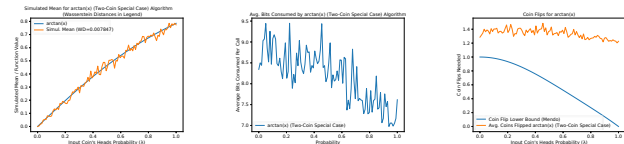
$\arcsin(x)/2$



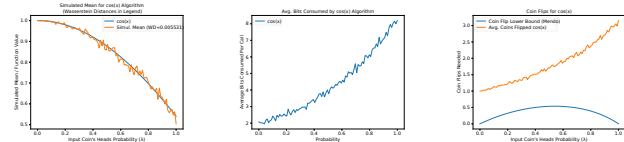
$\arctan(x)$   
(Flajolet)



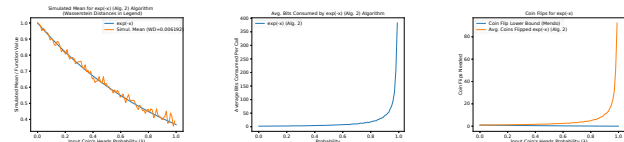
$\arctan(x)$  (Two-Coin Special Case)



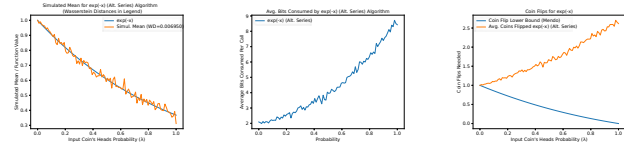
$\cos(x)$



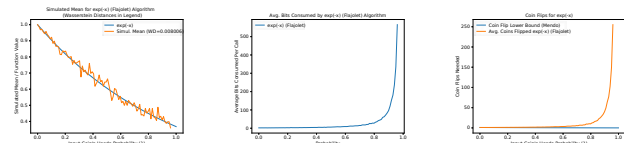
$\exp(-x)$  (Alg. 2)



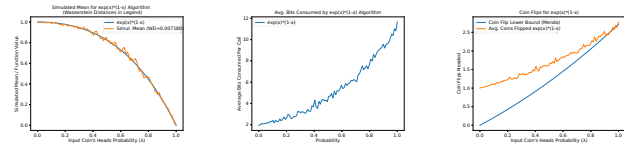
$\exp(-x)$  (Alt. Series)



$\exp(-x)$  (Flajolet)



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



[illegible]

Figure 1 consists of three subplots labeled (a), (b), and (c), each showing the performance of the proposed algorithm (blue line) compared to the Lasso method (orange line) across different values of the regularization parameter  $\lambda$  (ranging from 0.0 to 1.0).

- (a) Normalized MSE:** The y-axis is 'Normalized MSE (10<sup>-3</sup>)' ranging from 0.0 to 1.0. The x-axis is ' $\lambda$ ' ranging from 0.0 to 1.0. The proposed algorithm (blue line) shows a decreasing trend in MSE as  $\lambda$  increases, while the Lasso method (orange line) shows an increasing trend. The proposed algorithm's MSE is consistently lower than the Lasso method's MSE.
- (b) Normalized RMSE:** The y-axis is 'Normalized RMSE (10<sup>-3</sup>)' ranging from 0.0 to 1.0. The x-axis is ' $\lambda$ ' ranging from 0.0 to 1.0. The proposed algorithm (blue line) shows a decreasing trend in RMSE as  $\lambda$  increases, while the Lasso method (orange line) shows an increasing trend. The proposed algorithm's RMSE is consistently lower than the Lasso method's RMSE.
- (c) Coefficient Sparsity:** The y-axis is 'Coeff. Sparsity' ranging from 0.0 to 1.0. The x-axis is ' $\lambda$ ' ranging from 0.0 to 1.0. The proposed algorithm (blue line) shows a decreasing trend in coefficient sparsity as  $\lambda$  increases, while the Lasso method (orange line) shows an increasing trend. The proposed algorithm's coefficient sparsity is consistently lower than the Lasso method's coefficient sparsity.

[illegible]

Figure 10 consists of three subplots showing the performance of the proposed algorithm compared to the baseline algorithm. The left plot shows the reconstruction error (RMSE) for the proposed algorithm (blue line) and the baseline algorithm (orange line) as a function of input Cor'n's sparsity (0.0 to 1.0). The middle plot shows the RMSE for the proposed algorithm (blue line) and the baseline algorithm (orange line) as a function of input Cor'n's sparsity (0.0 to 1.0). The right plot shows the RMSE for the proposed algorithm (blue line) and the baseline algorithm (orange line) as a function of input Cor'n's sparsity (0.0 to 1.0).

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 Input Cori's records (0.0-1.0)'. It shows a red line for 'Actual, Mean (0.9141-0.9388)' and a blue line for 'Proposed, Mean (0.9141-0.9388)'. Both lines are nearly identical, starting at (0,0) and ending at (1,1).
- (b) Mean time consumed by parallel algorithm against probability:** The x-axis is 'Probability' (0.0 to 1.0) and the y-axis is 'Mean Time Consumed (Sec) (0-40)'. A blue line labeled 'Proposed, 100' starts at approximately 38 seconds for probability 0.0 and drops sharply to near zero by probability 0.1, remaining flat thereafter.
- (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 (0-140)'. It shows a blue line for 'Actual, Cori flag (0.9141-0.9388)' and an orange line for 'Proposed, Cori flag (0.9141-0.9388)'. Both lines start at approximately 135 for probability 0.0 and drop sharply to near zero by probability 0.1, remaining flat thereafter.

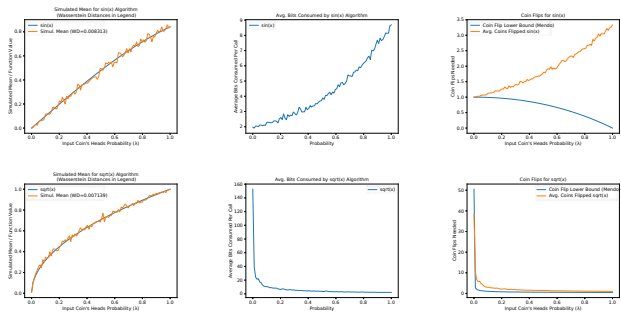
Figure 1 consists of three subplots. Subplot (a) is a line graph titled 'Regression Results for power\_AE2'. The x-axis is 'Input Carlo's memory Probability (x)' ranging from 0.0 to 1.0. The y-axis is 'Regression Results for power\_AE2' ranging from 0.0 to 1.0. It shows two data series: 'power\_AE2' (blue line with circles) and 'Linear Model (Power=0.000000)' (orange line). Both series show a linear increase from (0,0) to (1,1). Subplot (b) is a line graph titled 'Avg. CPU Consumed by power\_AE2 Algorithm'. The x-axis is 'Probability' ranging from 0.0 to 1.0. The y-axis is 'Average CPU Time (sec)' ranging from 0 to 15. The data series 'power\_AE2' (blue line with circles) shows a sharp decrease in CPU time from approximately 14 seconds at probability 0.0 to near 0 seconds at probability 1.0. Subplot (c) is a line graph titled 'CPU Time for power\_AE2'. The x-axis is 'Input Carlo's memory Probability (x)' ranging from 0.0 to 1.0. The y-axis is 'CPU Time (sec)' ranging from 0 to 4. It shows two data series: 'CPU Time Consumed (Power)' (blue line with circles) and 'Avg. CPU Time for power\_AE2' (orange line). Both series show a sharp decrease in CPU time from approximately 4 seconds at probability 0.0 to near 0 seconds at probability 1.0.

Figure 1 consists of four plots arranged in a 2x2 grid, showing the performance of different methods across various input correlation's feature probabilities (0.0 to 1.0).

- Top Left Plot:** Discretized Area Under the Curve (Y-axis, 0.0 to 1.0) vs. Input Correlation's Feature Probability (X-axis, 0.0 to 1.0). The plot shows two curves: a red curve for 'method G3 (20)' and a blue curve for 'method G3 (20) + method G3 (20)'. Both curves start at 0.0 and increase sharply after 0.5, reaching 1.0 at 1.0.
- Top Right Plot:** Average Area Under the Curve (Y-axis, 0.0 to 1.0) vs. Probability (X-axis, 0.0 to 1.0). The plot shows two curves: a red curve for 'method G3 (20)' and a blue curve for 'method G3 (20) + method G3 (20)'. Both curves start at 0.0 and increase sharply after 0.5, reaching 1.0 at 1.0.
- Bottom Left Plot:** Average Area Under the Curve (Y-axis, 0.0 to 1.0) vs. Probability (X-axis, 0.0 to 1.0). The plot shows two curves: a red curve for 'method G3 (20)' and a blue curve for 'method G3 (20) + method G3 (20)'. Both curves start at 0.0 and increase sharply after 0.5, reaching 1.0 at 1.0.
- Bottom Right Plot:** CDF of the Error (Y-axis, 0.0 to 1.0) vs. Input Correlation's Feature Probability (X-axis, 0.0 to 1.0). The plot shows two curves: a red curve for 'method G3 (20)' and a blue curve for 'method G3 (20) + method G3 (20)'. Both curves start at 0.0 and increase sharply after 0.5, reaching 1.0 at 1.0.

Figure 10 consists of three subplots. The left plot shows the Standard Error (Y-axis, 0.0 to 0.4) versus Input data's frequency (Hz) (X-axis, 0.0 to 1.0) for power 5.0 W. It compares the proposed model (blue line) with the proposed model without delay (orange line). Both lines show a linear increase, with the proposed model having a slightly lower error. The middle plot shows the Signal-to-Noise Ratio (SNR) (Y-axis, 0.0 to 1.0) versus Frequency (X-axis, 0.0 to 1.0) for power 5.0 W. The SNR drops sharply from 1.0 to 0.0 at a frequency of approximately 0.1 Hz. The right plot shows the Cost Function (Y-axis, 0.0 to 1000) versus Input data's frequency (Hz) (X-axis, 0.0 to 1.0) for power 5.0 W. The cost function drops sharply from 1000 to 0.0 at a frequency of approximately 0.1 Hz.

$\sin(x)$



$\sqrt{x}$

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