

Correctness and Performance Charts

Peter Occil

1 Correctness and Performance Charts

This version of the document is dated 2025-09-30.

The following charts show the correctness of many of the algorithms in “[Bernoulli Factory Algorithms¹](#)” 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.

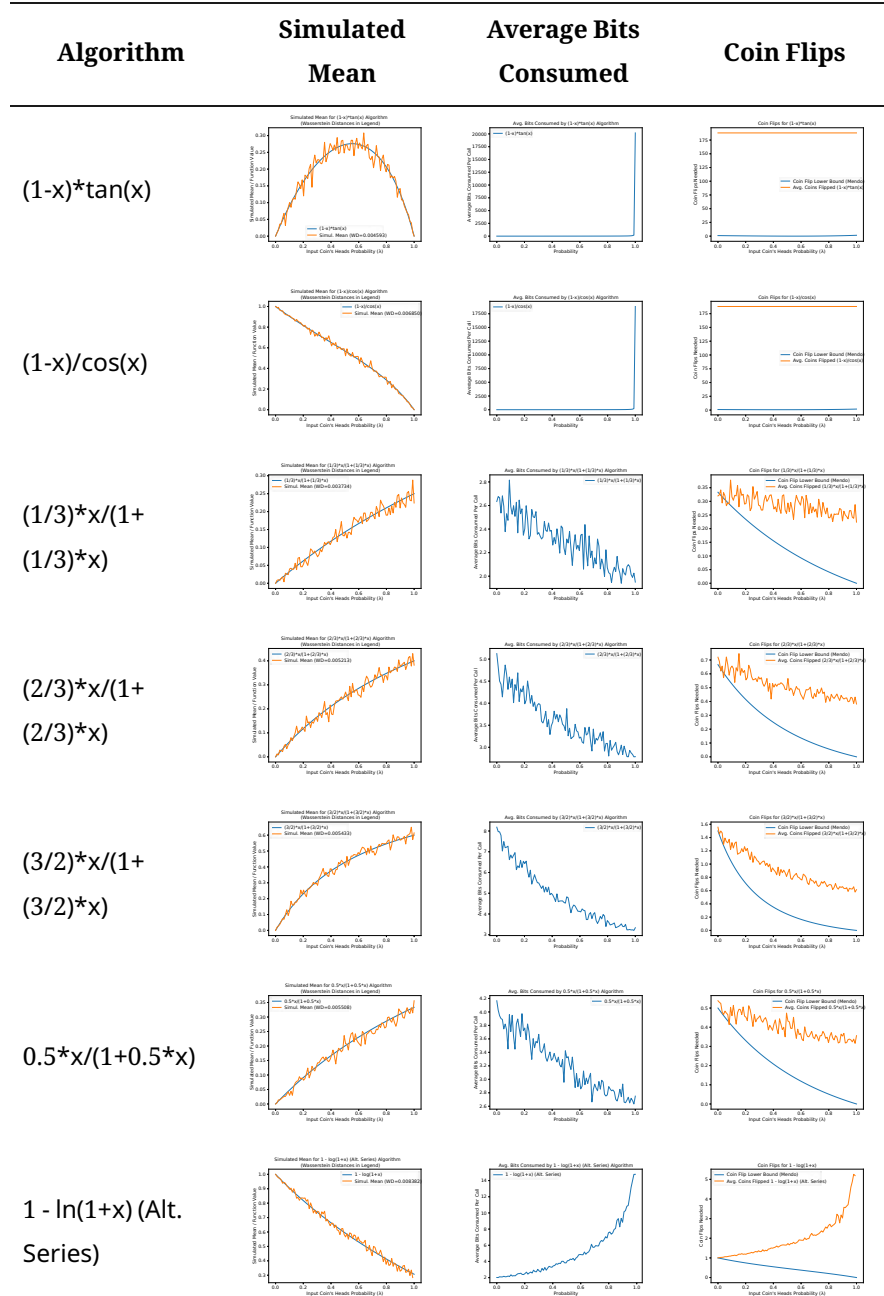
For each algorithm, if a single run was detected to use more than 5000 bits for a given λ , the entire data point for that λ was suppressed in the charts below.

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—

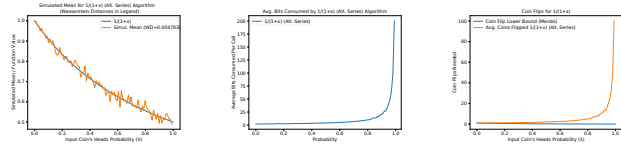
- an ϵ of $1 - (x + c) * 1.001$ was used (or 0.0001 if ϵ would be greater than 1), and
- an ϵ of $(x - c) * 0.9995$ for the subtraction variants.

Points with invalid ϵ values were suppressed. For the low-mean algorithm, an m of $\max(0.49999, x \cdot 1.02)$ was used unless noted otherwise.

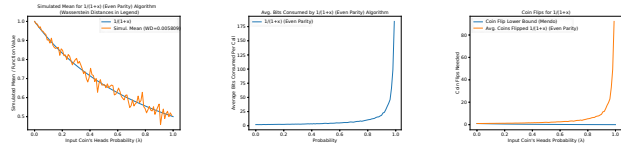
1.1 The Charts



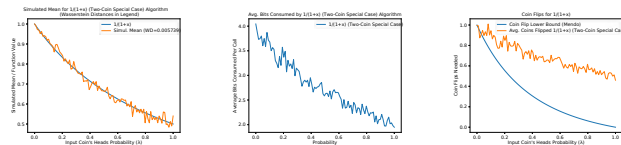
$1/(1+x)$ (Alt.
Series)



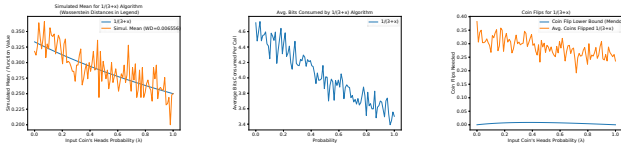
$1/(1+x)$ (Even
Parity)



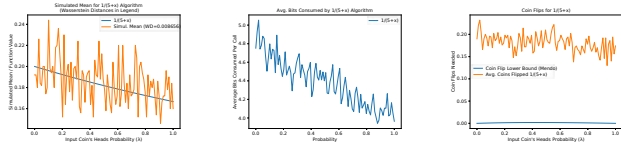
$1/(1+x)$ (Two-
Coin Special
Case)



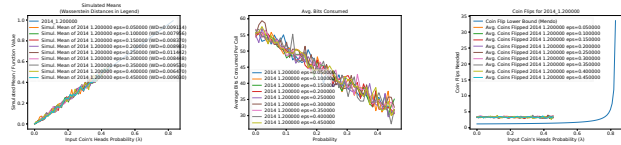
$1/(3+x)$



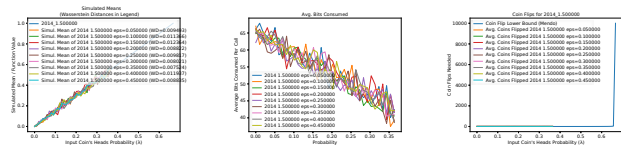
$1/(5+x)$



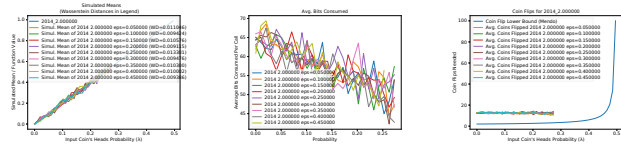
2014 1.200000
eps=0.050000



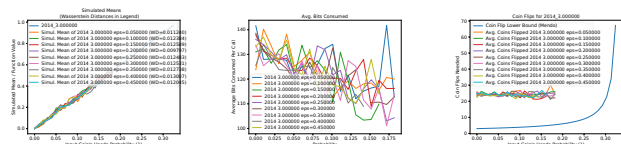
2014 1.500000
eps=0.050000



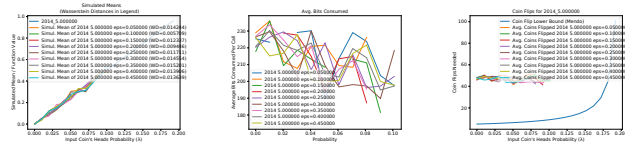
2014 2.000000
eps=0.050000



2014 3.000000
eps=0.050000



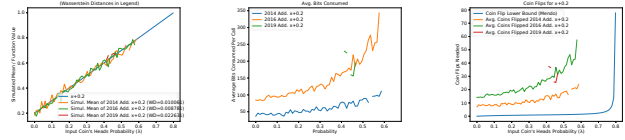
2014 5.000000
eps=0.050000



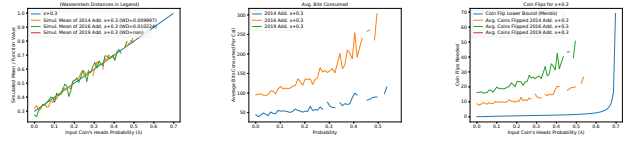
2014 Add. x+0.1



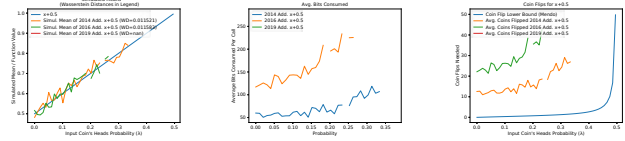
2014 Add. x+0.2



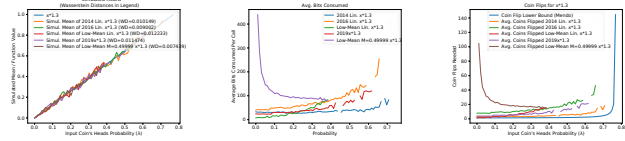
2014 Add. x+0.3



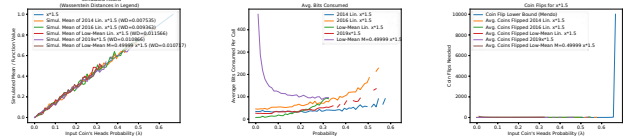
2014 Add. x+0.5



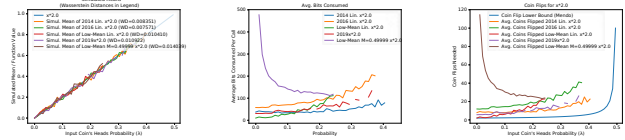
2014 Lin. x*1.3



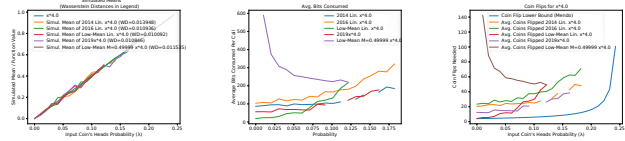
2014 Lin. x*1.5



2014 Lin. x*2.0



2014 Lin. x*4.0



[illegible]

Figure 1 consists of three subplots comparing the proposed model (solid lines) with the baseline model (dashed lines) across various datasets and models. The datasets include 2013-14, 2014-15, 2015-16, 2016-17, 2017-18, 2018-19, and 2019-20. The models compared are Logistic Regression, Random Forest, and XGBoost.

(a) ROC Curve: The y-axis is True Positive Rate (0.0 to 1.0) and the x-axis is False Positive Rate (0.0 to 1.0). The proposed model (solid lines) shows a higher True Positive Rate for a given False Positive Rate compared to the baseline model (dashed lines).

(b) AUC vs. Probability: The y-axis is AUC (0.0 to 1.0) and the x-axis is Probability (0.0 to 1.0). The proposed model (solid lines) shows a higher AUC for a given Probability compared to the baseline model (dashed lines).

(c) C-index: The y-axis is C-index (0.0 to 1.0) and the x-axis is Probability (0.0 to 1.0). The proposed model (solid lines) shows a higher C-index for a given Probability compared to the baseline model (dashed lines).

Figure 1 consists of three subplots. Subplot (a) is a scatter plot titled 'Scatterplot Results' showing 'True Labels' on the y-axis and 'Ground Labels' on the x-axis, both ranging from 0.0 to 0.6. A diagonal line represents perfect agreement. Data points are clustered along this line, with a legend indicating '2014_1_2015'. Subplot (b) is a line plot titled 'Avg. RMSE Comparison' showing 'Avg. RMSE' on the y-axis (0.0 to 0.4) versus 'Ground Labels' on the x-axis (0.0 to 0.6). It contains multiple colored lines representing different models and years (2014 and 2015), with a legend on the right. Subplot (c) is a confusion matrix titled 'Conf. Matrix Results' for 'Conf. Matrix Results 2014_1_2015'. The x-axis is 'Actual' and the y-axis is 'Predicted', both with categories '0' and '1'. The matrix shows counts for true positives, false positives, false negatives, and true negatives.

[illegible]

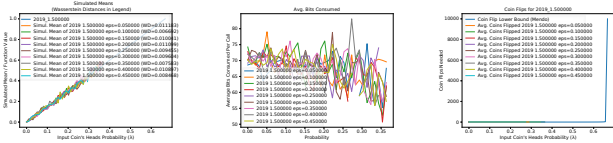
Figure 1 consists of three subplots labeled (a), (b), and (c).
 Subplot (a) is a scatter plot titled "Scatter Plot Results (Random Samples = 10000)". The x-axis is labeled "% True Accurate" and the y-axis is labeled "% Accurate". Both axes range from 0.0 to 1.0. A diagonal line from (0,0) to (1,1) is shown. Data points for 10 different runs are plotted, showing a strong positive correlation. A legend lists the runs with their corresponding accuracy values.
 Subplot (b) is a line plot titled "Avg. Cos. Similarity (Random Samples = 10000)". The x-axis is labeled "Average Cosine Similarity" and the y-axis is labeled "Average Cosine Similarity". Both axes range from 0.00 to 1.00. Multiple colored lines represent different runs, showing fluctuations between 0.5 and 1.0. A legend lists the runs with their corresponding average cosine similarity values.
 Subplot (c) is a confusion matrix titled "Conf. Mat. Results for 10000 Random Samples". The x-axis is labeled "Actual Class" and the y-axis is labeled "Conf. Mat. Results". Both axes have categories: "True", "False", "Both", and "None". The matrix shows counts for each combination of actual and predicted class. A legend lists the runs with their corresponding confusion matrix results.

Figure 1 consists of three subplots. Subplot (a) is a scatter plot titled 'Dependent Variable vs. Independent Variable' showing the relationship between 'True Value' (y-axis) and 'Predicted Value' (x-axis). The data points are clustered around the diagonal line, indicating high predictive accuracy. Subplot (b) is a Receiver Operating Characteristic (ROC) curve showing the True Positive Rate (y-axis) versus the False Positive Rate (x-axis). The curve is significantly above the diagonal line, indicating good model performance. Subplot (c) is a Confusion Matrix showing the counts of True Positives, True Negatives, False Positives, and False Negatives. The matrix is a 2x2 grid with values ranging from 0 to 1000.

Figure 1 consists of three subplots. Subplot (a) is titled "Performance Results" and shows ROC curves for 10 datasets: 2016, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, and 2008. The x-axis is "True Positive Rate" and the y-axis is "True Negative Rate", both ranging from 0.00 to 1.00. Subplot (b) is titled "Avg. ROC Curves" and shows the average ROC curves for sample sizes of 2000, 4000, 6000, 8000, 10000, 12000, 14000, 16000, 18000, and 20000. The x-axis is "True Positive Rate" and the y-axis is "True Negative Rate", both ranging from 0.00 to 1.00. Subplot (c) is titled "Confusion Matrix" and shows the confusion matrix for the 2016 dataset with 10,000 samples. The x-axis is "Actual" and the y-axis is "Predicted", both with categories "0" and "1". The matrix shows counts for true positives, true negatives, false positives, and false negatives.

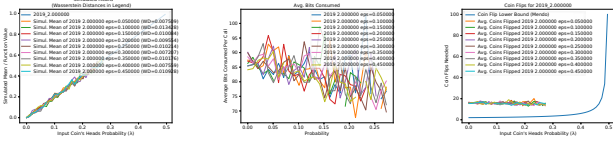
Figure 1 consists of three subplots. The left subplot, titled 'Scatterplot Results', shows 'True Cases Number' on the y-axis (0 to 1000) versus 'Predicted Cases Number' on the x-axis (0 to 1000). It contains multiple data series for different dates from 2020-03-15 to 2020-04-20, with points generally following the diagonal line. The middle subplot, titled 'Avg. Day Comparison', shows 'Average Day Comparison Coefficient' on the y-axis (0.8 to 1.0) versus 'date' on the x-axis (2020-03-15 to 2020-04-20). It displays multiple lines for different dates, mostly fluctuating between 0.8 and 1.0. The right subplot, titled 'Correlation Results', shows 'Correlation Coefficient' on the y-axis (0.0 to 1.0) versus 'date' on the x-axis (2020-03-15 to 2020-04-20). It shows multiple lines for different dates, with most lines near 1.0 and one line (2020-04-20) dropping to 0.0.

eps=0.050000



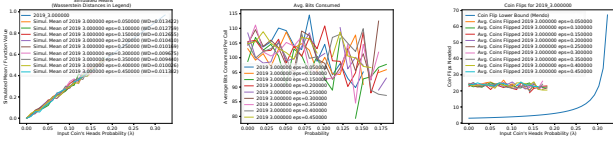
2019 2.000000

eps=0.050000



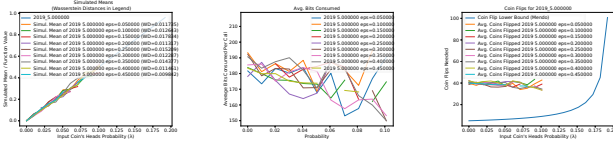
2019 3.000000

eps=0.050000



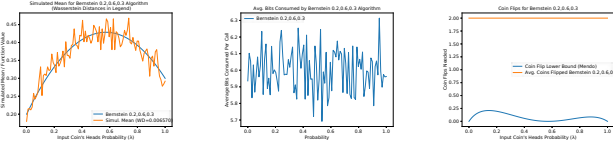
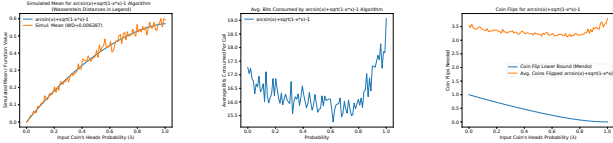
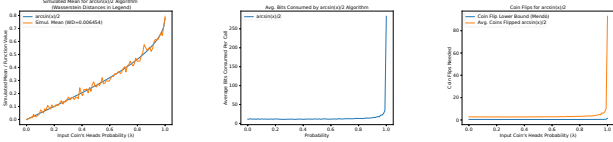
2019 5.000000

eps=0.050000

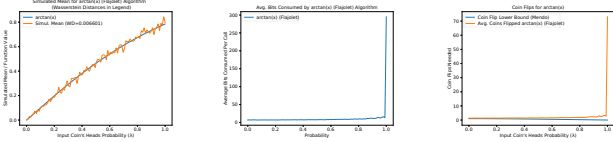


Bernstein

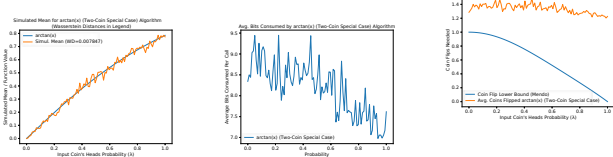
0.2,0.6,0.3


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

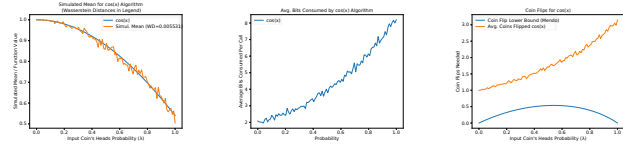
(Flajolet)



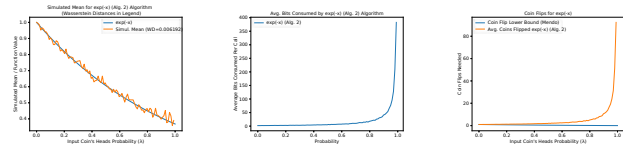
arctan(x) (Two-Coin Special Case)



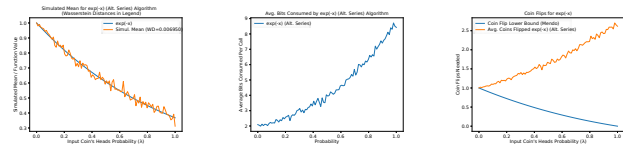
$\cos(x)$



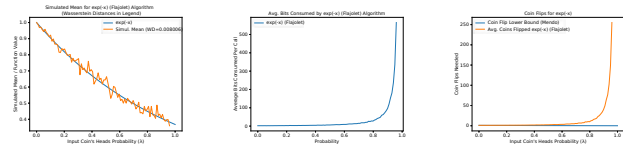
$\exp(-x)$ (Alg. 2)



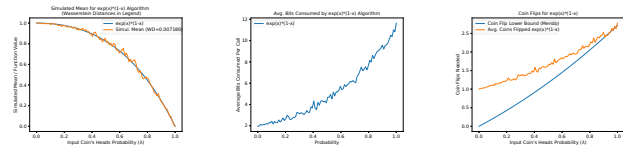
$\exp(-x)$ (Alt. Series)



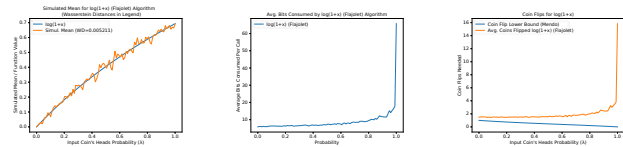
$\exp(-x)$ (Flajolet)



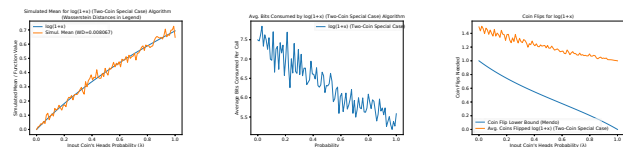
$\exp(x)*(1-x)$



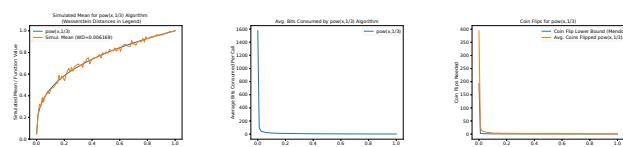
$\ln(1+x)$ (Flajolet)



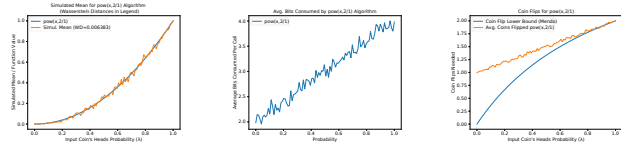
$\ln(1+x)$ (Two-Coin Special Case)



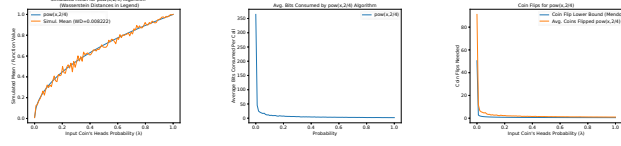
$\text{pow}(x, 1/3)$



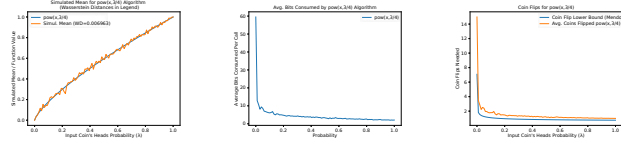
pow(x,2/1)



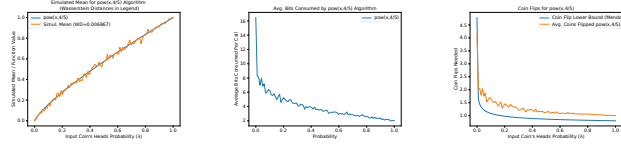
pow(x,2/4)



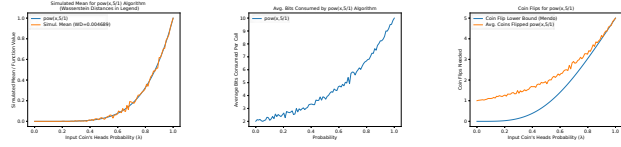
pow(x,3/4)



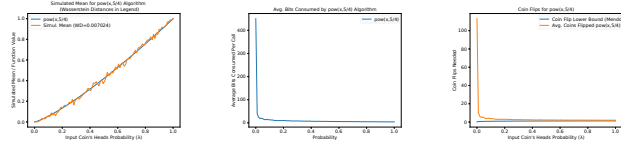
pow(x,4/5)



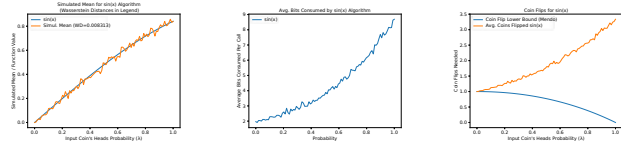
pow(x,5/1)



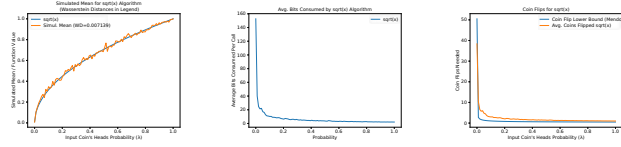
pow(x,5/4)



sin(x)



sqrt(x)



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