

## # Correctness and Performance Charts

This version of the document is dated 2022-11-07.

The following charts show the correctness of many of the algorithms in "**Bernoulli Factory Algorithms**" 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 given 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—

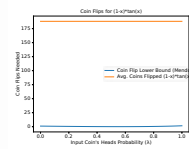
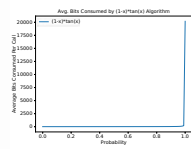
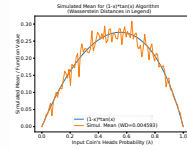
- an  $\epsilon$  of  $1 - (x + c) * 1.001$  was used (or 0.0001 if  $\epsilon$  would be greater than 1), and
- an  $\epsilon$  of  $(x - c) * 0.9995$  for the subtraction variants.

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

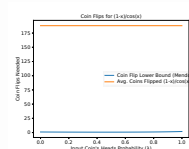
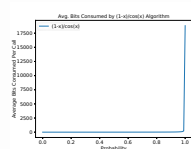
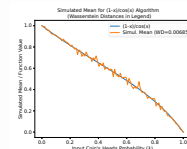
## 0.1 The Charts

Algorithm	Simulated Mean	Average Bits Consumed	Coin Flips
-----------	----------------	-----------------------	------------

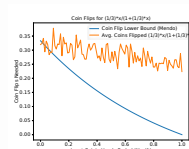
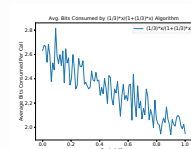
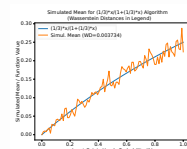
$$(1-x)*\tan(x)$$



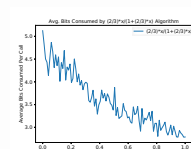
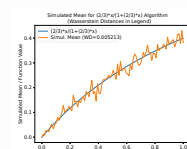
$$(1-x)/\cos(x)$$



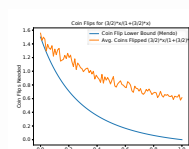
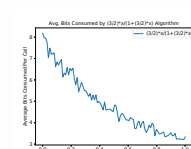
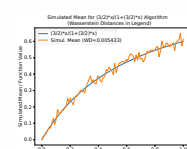
$$(1/3)*x/(1+(1/3)*x)$$



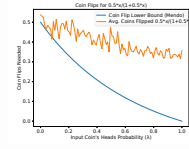
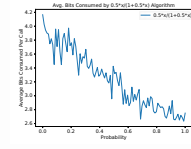
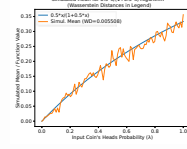
$$(2/3)*x/(1+(2/3)*x)$$



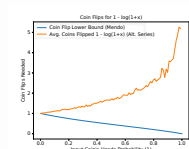
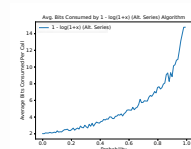
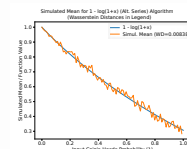
$$(3/2)*x/(1+(3/2)*x)$$



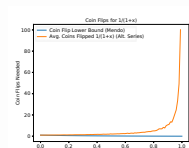
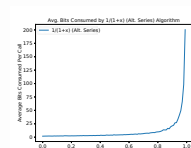
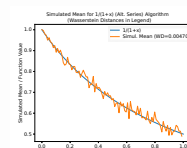
$$0.5*x/(1+0.5*x)$$



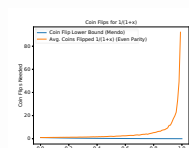
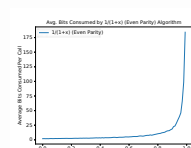
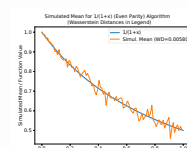
$$1 - \ln(1+x) \text{ (Alt. Series)}$$



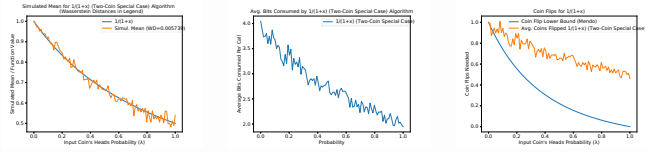
$$1/(1+x) \text{ (Alt. Series)}$$



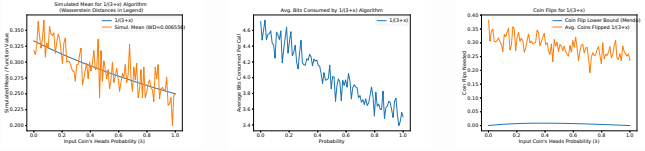
$$1/(1+x) \text{ (Even Parity)}$$



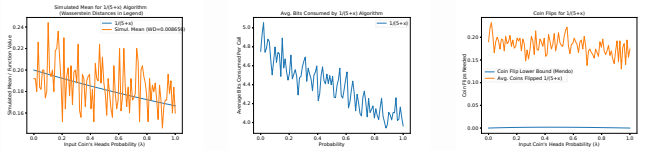
1/(1+x) (Two-Coin Special Case)



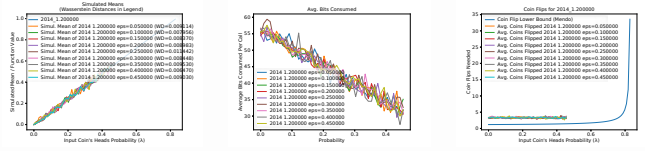
1/(3+x)



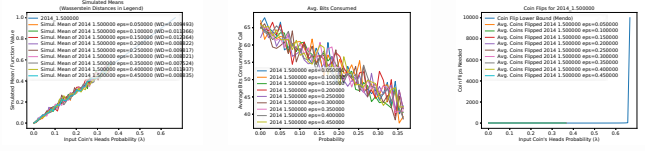
1/(5+x)



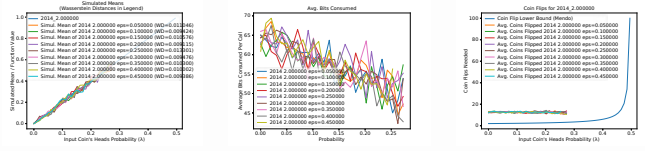
2014 1.200000  
eps=0.050000



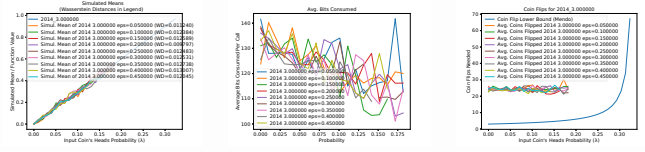
2014 1.500000  
eps=0.050000



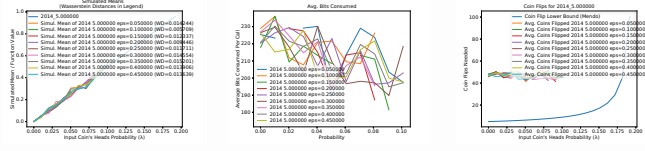
2014 2.000000  
eps=0.050000



2014 3.000000  
eps=0.050000



2014 5.000000  
eps=0.050000



2014 Add. x+0.1

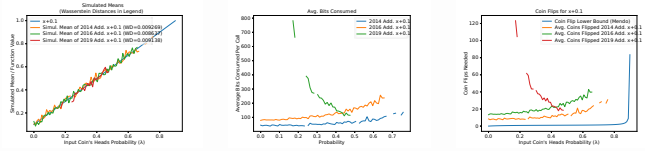


Figure 1 consists of three subplots. The left plot, titled 'Disaggregated Rain (Observations minus Forecast)', shows 'True Rain (mm)' on the y-axis (0.0 to 1.0) versus 'Predicted Rain (mm)' on the x-axis (0.0 to 1.0). It includes a blue diagonal line for perfect agreement and data points for 2014 (green), 2015 (orange), and 2016 (red). The middle plot, titled 'Avg. RMSE Computed', shows 'Average RMSE (mm)' on the y-axis (0.0 to 0.05) versus 'Normalized Error' on the x-axis (0.0 to 0.6). It displays three lines for 2014 (green), 2015 (orange), and 2016 (red). The right plot, titled 'Error Plots for 2017-2019', shows 'RMSE (mm)' on the y-axis (0 to 80) versus 'Normalized Error' on the x-axis (0.0 to 0.7). It displays three lines for 2017 (green), 2018 (orange), and 2019 (red).

Figure 1 consists of three subplots. The top-left plot, titled 'Comparison of the proposed method with the state-of-the-art', shows the 'Stratified Mean AUC (std)' for various models and datasets. The x-axis ranges from 0.6 to 0.8, and the y-axis ranges from 0.6 to 0.8. The legend indicates: 'Proposed' (blue line), '2018 AAAI' (orange line), '2018 AAAI' (green line), '2018 AAAI' (red line), and '2018 AAAI' (purple line). The top-right plot, titled 'Average ROC Curves for AUC', shows the 'Average ROC Curves for AUC' for the same models and datasets. The x-axis ranges from 0.6 to 0.8, and the y-axis ranges from 0.6 to 0.8. The bottom plot, titled 'CUP Filter for AUC=0.5', shows the 'CUP Filter for AUC=0.5' for the same models and datasets. The x-axis ranges from 0.6 to 0.8, and the y-axis ranges from 0.6 to 0.8.

[illegible]

Figure 1 consists of three subplots. Subplot (a) shows ROC curves for six datasets: C5.0, C4.5, C4.5, C4.5, C4.5, and C4.5. The x-axis is 'Input Cost Ratio (Probability)' and the y-axis is 'Recall Rate (Cost Ratio)'. Subplot (b) shows Average ROC Curves for v1.5, with the x-axis being 'Input Cost Ratio (Probability)' and the y-axis being 'Average ROC Curves for v1.5'. Subplot (c) shows the Cost-Peak Ratio for v1.5, with the x-axis being 'Input Cost Ratio (Probability)' and the y-axis being 'Cost-Peak Ratio'.

Figure 1 consists of three subplots labeled (a), (b), and (c).  
 Subplot (a) is a scatter plot titled "Scatter Plot of Predicted vs. Actual Mean (Covariance)". The x-axis is "Input C<sub>1</sub> Mean Probability (%)" ranging from 0.0 to 0.5. The y-axis is "Predicted Mean (Covariance)" ranging from 0.0 to 1.0. Data points are colored circles representing different input mean probabilities: 0.0, 0.1, 0.2, 0.3, 0.4, and 0.5. A red diagonal line represents the ideal prediction. The points closely follow this line.  
 Subplot (b) is a line plot titled "Avg. Mean Squared Error (MSE) vs. Probability". The x-axis is "Probability" ranging from 0.0 to 0.5. The y-axis is "Average Mean Squared Error (MSE)" ranging from 0.0 to 0.05. Multiple lines represent different input mean probabilities: 0.0 (blue), 0.1 (orange), 0.2 (green), 0.3 (red), 0.4 (purple), and 0.5 (brown). The MSE generally increases with probability, with the 0.5 input mean probability showing the highest error.  
 Subplot (c) is a line plot titled "Cost Ratio vs. Probability". The x-axis is "Input C<sub>1</sub> Mean Probability (%)" ranging from 0.0 to 0.5. The y-axis is "Cost Ratio" ranging from 0 to 100. Multiple lines represent different input mean probabilities: 0.0 (blue), 0.1 (orange), 0.2 (green), 0.3 (red), 0.4 (purple), and 0.5 (brown). The cost ratio is highest at low probabilities and decreases as probability increases, with the 0.5 input mean probability showing the lowest cost ratio.

Figure 1 consists of two subplots, (a) and (b), showing the performance of the proposed model.

Subplot (a) is a Receiver Operating Characteristic (ROC) curve. The x-axis is labeled "Input Coir's Weave Probability (x)" and ranges from 0.00 to 0.40. The y-axis is labeled "Detection Rate (True Positive Rate)" and ranges from 0.00 to 1.00. The plot shows several curves for different models:
 

- Proposed (blue line): This curve is the highest, indicating the best performance.
- Proposed + 2018-2019 (orange line): This curve is slightly below the proposed model.
- Proposed + 2018-2019 + 2020-2021 (green line): This curve is below the previous one.
- Proposed + 2018-2019 + 2020-2021 + 2022-2023 (red line): This curve is below the previous one.
- Proposed + 2018-2019 + 2020-2021 + 2022-2023 + 2024-2025 (purple line): This curve is the lowest among the proposed model variants.
- Random (black line): This is a diagonal line from (0,0) to (1,1), representing a random classifier.

Subplot (b) is an Average ROC Curve for the proposed model. The x-axis is labeled "Input Coir's Weave Probability (x)" and ranges from 0.00 to 0.40. The y-axis is labeled "Average ROC Curve" and ranges from 0.00 to 1.00. The plot shows several curves for different input data sets:
 

- Coir Weave (blue line): This curve is the highest, indicating the best performance.
- Coir Weave + 2018-2019 (orange line): This curve is slightly below the Coir Weave model.
- Coir Weave + 2018-2019 + 2020-2021 (green line): This curve is below the previous one.
- Coir Weave + 2018-2019 + 2020-2021 + 2022-2023 (red line): This curve is below the previous one.
- Coir Weave + 2018-2019 + 2020-2021 + 2022-2023 + 2024-2025 (purple line): This curve is the lowest among the proposed model variants.
- Random (black line): This is a diagonal line from (0,0) to (1,1), representing a random classifier.

The figure consists of three vertically stacked plots sharing a common x-axis representing input correlation probability from 0.00 to 0.15.

- Top Plot:** Average Normalized Error Rate vs. Input Correlation Probability. The y-axis ranges from 0.00 to 1.00. All methods show an increasing error rate as correlation increases. The proposed method (blue) generally has the lowest error rate across most correlations.
- Middle Plot:** Average Coverage Rate (%) vs. Probability. The y-axis ranges from 0 to 100. Most methods maintain coverage near 100%, while the proposed method (blue) starts lower at ~80% and decreases slightly as correlation increases.
- Bottom Plot:** Cost per Bit vs. Input Correlation Probability. The y-axis ranges from 0 to 160. The proposed method (blue) shows a sharp increase in cost starting around 0.10 correlation, reaching over 160 at 0.15 correlation. Other methods remain relatively flat or show much less dramatic increases.

Figure 1 consists of two plots, (a) and (b), comparing the proposed model with existing models.

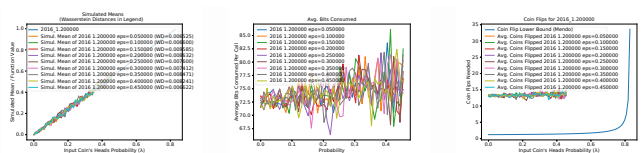
Plot (a) shows the Standard Error of the Mean (SEM) on the y-axis (ranging from 0.00 to 0.02) versus Input Concentration (mg/L) on the x-axis (ranging from 0.00 to 0.60). The legend includes:
 

- Prop. (red line)
- Linear (blue line)
- Linear + 1st Order (green line)
- Linear + 2nd Order (purple line)
- Linear + 3rd Order (orange line)
- Linear + 4th Order (brown line)
- Linear + 5th Order (pink line)
- Linear + 6th Order (grey line)
- Linear + 7th Order (light blue line)
- Linear + 8th Order (light green line)
- Linear + 9th Order (light orange line)
- Linear + 10th Order (light purple line)
- Linear + 11th Order (light brown line)
- Linear + 12th Order (light pink line)
- Linear + 13th Order (light grey line)
- Linear + 14th Order (light blue line)
- Linear + 15th Order (light green line)
- Linear + 16th Order (light orange line)
- Linear + 17th Order (light purple line)
- Linear + 18th Order (light brown line)
- Linear + 19th Order (light pink line)
- Linear + 20th Order (light grey line)

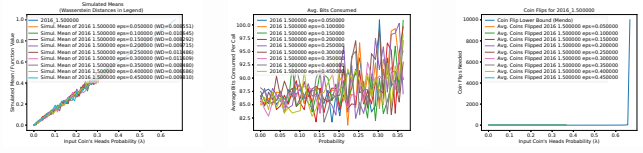
Plot (b) shows the Average Bio-Concentration Factor (BCF) on the y-axis (ranging from 0 to 200) versus Input Concentration (mg/L) on the x-axis (ranging from 0.00 to 0.60). The legend includes:
 

- Prop. (red line)
- Linear (blue line)
- Linear + 1st Order (green line)
- Linear + 2nd Order (purple line)
- Linear + 3rd Order (orange line)
- Linear + 4th Order (brown line)
- Linear + 5th Order (pink line)
- Linear + 6th Order (grey line)
- Linear + 7th Order (light blue line)
- Linear + 8th Order (light green line)
- Linear + 9th Order (light orange line)
- Linear + 10th Order (light purple line)
- Linear + 11th Order (light brown line)
- Linear + 12th Order (light pink line)
- Linear + 13th Order (light grey line)
- Linear + 14th Order (light blue line)
- Linear + 15th Order (light green line)
- Linear + 16th Order (light orange line)
- Linear + 17th Order (light purple line)
- Linear + 18th Order (light brown line)
- Linear + 19th Order (light pink line)
- Linear + 20th Order (light grey line)

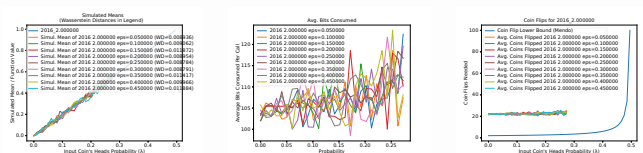
```
2016 1.200000
eps=0.050000
```



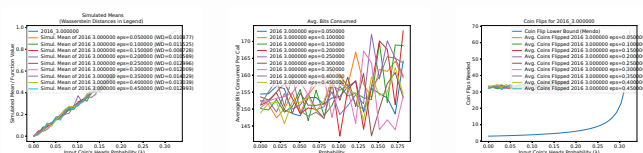
```
2016 1.500000
eps=0.050000
```



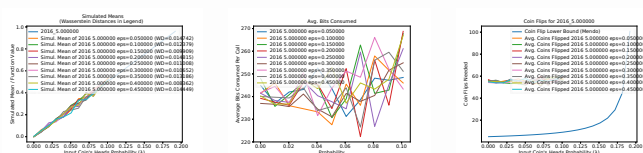
```
2016 2.000000
eps=0.050000
```



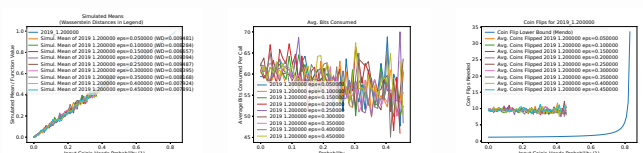
```
2016 3.000000
eps=0.050000
```



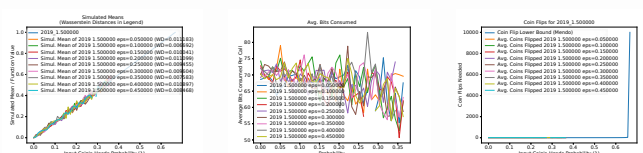
```
2016 5.000000
eps=0.050000
```



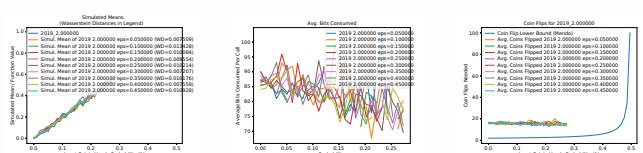
```
2019 1.200000
eps=0.050000
```



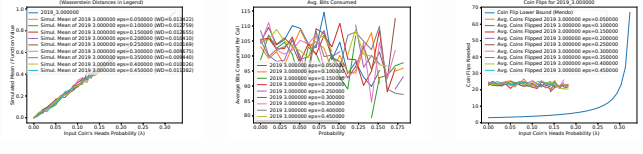
```
2019 1.500000
eps=0.050000
```



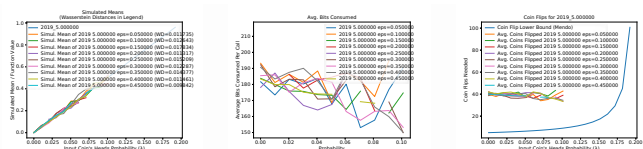
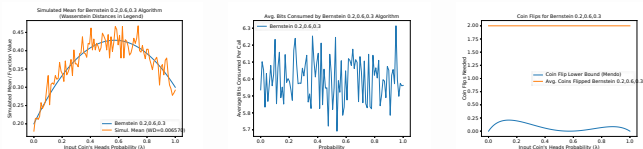
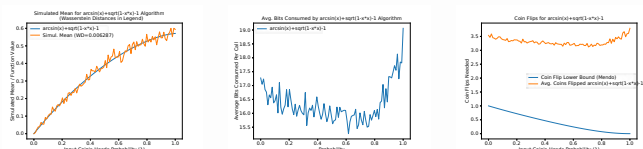
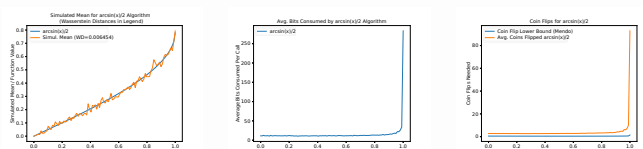
```
2019 2.000000
eps=0.050000
```



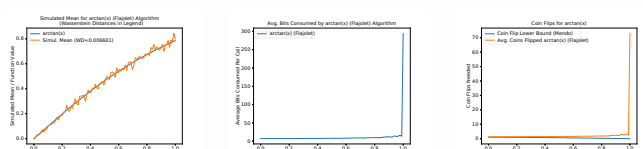
```
2019 3.000000
eps=0.050000
```



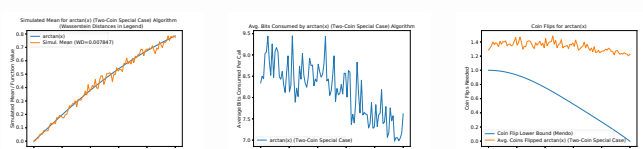
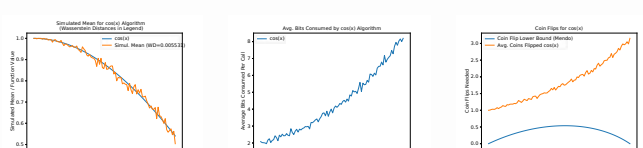
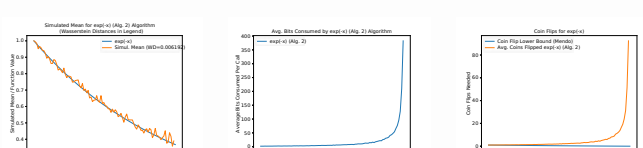
```
2019 5.000000
eps=0.050000
```

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)

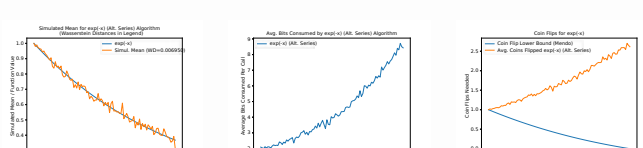


Figure 1 consists of three subplots labeled (a), (b), and (c), each showing performance metrics against the normalized GCM iterations (x-axis, ranging from 0.0 to 1.0).

- (a) Scaled error vs. normalized GCM iterations:** The y-axis is 'Scaled error (arbitrary unit)' ranging from 0.0 to 1.0. The legend indicates 'exact' (blue line) and 'Simplified error (proposed algorithm)' (orange line with markers). Both curves show a decreasing trend, with the proposed algorithm's error closely following the exact error.
- (b) Average bit complexity vs. normalized GCM iterations:** The y-axis is 'Average bit complexity (bit)' ranging from 0.0 to 1.0. The legend indicates 'exact' (blue line) and 'approx of (proposed)' (orange line with markers). The exact complexity is a constant horizontal line at 1.0, while the proposed algorithm's complexity starts at 1.0 and decreases to approximately 0.4 at x=1.0.
- (c) GCM steps vs. normalized GCM iterations:** The y-axis is 'GCM steps (bit)' ranging from 0 to 250. The legend indicates 'Exact (proposed algorithm)' (blue line) and 'Approx (proposed algorithm)' (orange line with markers). The exact steps are a constant horizontal line at approximately 10. The proposed algorithm's steps start at approximately 10 and increase sharply to over 250 as the normalized GCM iterations approach 1.0.

Figure 10 consists of three subplots. The left subplot, titled 'Scalability of the proposed algorithm', shows the 'Scalability of the proposed algorithm' on the y-axis (ranging from 0.0 to 1.0) versus 'Input Data Size (Number of Nodes)' on the x-axis (ranging from 0.0 to 1.0). It compares 'Proposed' (blue line) and 'Baseline' (orange line) algorithms. Both show a decreasing trend, with the proposed algorithm maintaining higher scalability at larger input sizes. The middle subplot, titled 'Avg. Bits Consumed by the proposed algorithm', shows 'Avg. Bits Consumed (in bits)' on the y-axis (ranging from 0 to 10) versus 'Input Data Size (Number of Nodes)' on the x-axis (ranging from 0.0 to 1.0). It compares 'Proposed' (blue line) and 'Baseline' (orange line) algorithms. Both show an increasing trend, with the proposed algorithm consuming fewer bits. The right subplot, titled 'Gap Error for input data size', shows 'Gap Error (in bits)' on the y-axis (ranging from 0.0 to 3.0) versus 'Input Data Size (Number of Nodes)' on the x-axis (ranging from 0.0 to 1.0). It compares 'Proposed' (blue line) and 'Baseline' (orange line) algorithms. Both show an increasing trend, with the proposed algorithm having a lower gap error.

[illegible]

Figure 1 consists of three subplots labeled (a), (b), and (c), each showing performance metrics versus Signal-to-Noise Ratio (SNR) on the x-axis (ranging from 0.0 to 1.0).

- (a) Standard Error (Std) (dB):** The y-axis ranges from -10 to 0. The legend indicates:
  - Blue line:  $\log_2(x)$  (Then Covert Special Case Algorithm)
  - Orange line:  $\log_2(x)$  (Then Covert Special Case Algorithm)
  - Red line:  $\log_2(x)$  (Then Covert Special Case Algorithm)
 All three lines overlap and show a linear increase from approximately -10 dB at SNR 0.0 to 0 dB at SNR 1.0.
- (b) Average Bias (Percent) (%):** The y-axis ranges from -10 to 10. The legend indicates:
  - Blue line:  $\log_2(x)$  (Then Covert Special Case Algorithm)
  - Orange line:  $\log_2(x)$  (Then Covert Special Case Algorithm)
  - Red line:  $\log_2(x)$  (Then Covert Special Case Algorithm)
 All three lines overlap and show a decreasing trend from approximately 10% at SNR 0.0 to -10% at SNR 1.0.
- (c) Coverage Probability (Prob):** The y-axis ranges from 0.0 to 1.0. The legend indicates:
  - Blue line: Covert Prob Lower Bound (Theoret)
  - Orange line: Covert Prob Upper Bound (Theoret)
  - Red line: Covert Prob (Then Covert Special Case Algorithm)
 The blue line (lower bound) decreases from approximately 0.8 at SNR 0.0 to 0.0 at SNR 1.0. The orange line (upper bound) decreases from approximately 1.0 at SNR 0.0 to approximately 0.8 at SNR 1.0. The red line (proposed algorithm) starts at approximately 0.9 at SNR 0.0 and decreases to approximately 0.8 at SNR 1.0, staying between the theoretical bounds.

Figure 1 consists of three subplots illustrating the performance of the proposed algorithm. The left subplot shows the Simulated Mean Error (Std) versus Signal-to-Noise Ratio (dB) for points 1,2,3 (blue line) and a single-point model (orange line). The middle subplot shows the Average Bit Error Rate versus Frequency for points 1,2,3 (blue line). The right subplot shows the Cost (Flips per point, 1,2) versus Signal-to-Noise Ratio (dB) for points 1,2,3 (blue line) and a single-point model (orange line).

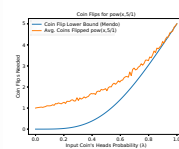
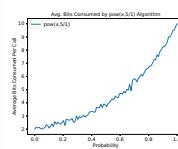
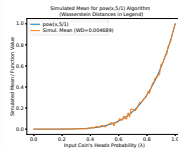
Figure 1 consists of three subplots, each showing an ROC curve. The x-axis for all plots is 'Input Gene's Feature Probability (x)' ranging from 0.0 to 1.0. The y-axis is 'Real Gene's Feature Probability (y)' ranging from 0.0 to 1.0. Each plot contains two lines: a blue line representing the proposed algorithm and an orange line representing a random classifier. The blue lines are consistently above the orange lines, indicating better performance.

- Simulated Data for gene 21's Algorithm:** The blue line starts at (0,0) and follows a smooth, upward curve, ending at (1,1). The orange line is a diagonal line from (0,0) to (1,1).
- Gene 21's Algorithm:** The blue line starts at (0,0) and follows a smooth, upward curve, ending at (1,1). The orange line is a diagonal line from (0,0) to (1,1).
- Gene 212's Algorithm:** The blue line starts at (0,0) and follows a smooth, upward curve, ending at (1,1). The orange line is a diagonal line from (0,0) to (1,1).

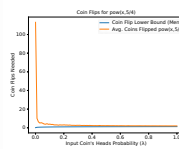
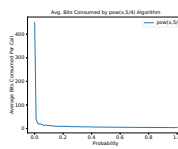
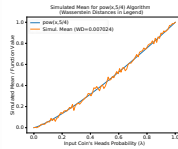
Figure 1 consists of three subplots. Subplot (a) is a line graph titled 'Dry weight after 1000 iterations' on the y-axis (ranging from 0.0 to 1.0) and 'Input Gene's Feature Probability (X)' on the x-axis (ranging from 0.0 to 1.0). It shows two data series: 'Proposed Algorithm' (blue line with circles) and 'Random Forest (1000-500000-50)' (orange line with circles). Both series show a similar increasing trend, starting near (0.0, 0.0) and ending near (1.0, 1.0). Subplot (b) is a line graph titled 'Area Under the Curve (AUC)' on the y-axis (ranging from 0.0 to 1.0) and 'Probability' on the x-axis (ranging from 0.0 to 1.0). It shows two data series: 'Proposed Algorithm' (blue line) and 'Random Forest' (orange line). The 'Proposed Algorithm' series starts at a high AUC of approximately 0.85 at probability 0.0 and drops sharply to near 0.0 by probability 0.1. The 'Random Forest' series starts at a high AUC of approximately 0.85 at probability 0.0 and remains relatively stable, ending near 0.8 at probability 1.0. Subplot (c) is a line graph titled 'Cost-Fit' on the y-axis (ranging from 0.0 to 1.6) and 'Input Gene's Feature Probability (X)' on the x-axis (ranging from 0.0 to 1.0). It shows two data series: 'Cost-Fit for the Proposed Algorithm' (blue line) and 'Cost-Fit for Random Forest' (orange line). Both series start at a high cost of approximately 1.5 at probability 0.0 and drop sharply to near 0.0 by probability 0.1. The 'Proposed Algorithm' series remains slightly higher than the 'Random Forest' series for the remainder of the probability range.

Figure 1 consists of three subplots. Subplot (a) is an ROC curve titled 'Stratified Mean for pnaive, A7C2 Algorithm' with 'Stratified Area Under the Curve (AUC)' on the y-axis and 'Input data's missing probability (%)' on the x-axis. It shows three curves: pnaive (blue), A7C2 (orange), and A7C3 (red), all increasing from 0.5 to 1.0. Subplot (b) is titled 'Avg. Size Computed by pnaive, A7C2, A7C3' with 'Average Size' on the y-axis and 'Probability' on the x-axis. It shows three curves: pnaive (blue), A7C2 (orange), and A7C3 (red), all decreasing from approximately 4.0 to 1.0. Subplot (c) is titled 'Avg. Time Computed by pnaive, A7C2, A7C3' with 'Average Time' on the y-axis and 'Input data's missing probability (%)' on the x-axis. It shows three curves: pnaive (blue), A7C2 (orange), and A7C3 (red), all decreasing from approximately 4.0 to 1.0.

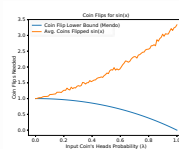
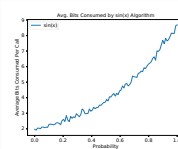
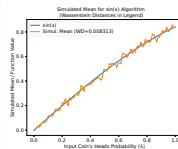
pow(x,5/1)



pow(x,5/4)



sin(x)



sqrt(x)

