

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 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 given 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—

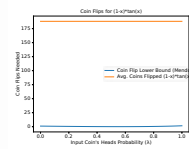
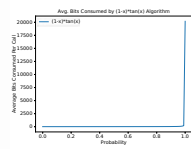
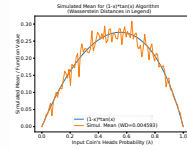
- an ϵ of $1 - (x + c) * 1.001$ was used (or 0.0001 if ϵ would be greater than 1), and
- an ϵ of $(x - c) * 0.9995$ for the subtraction variants.

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

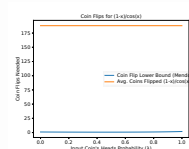
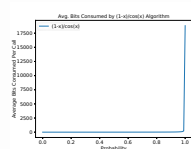
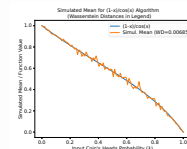
0.1 The Charts

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

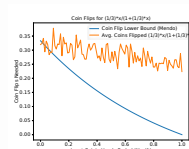
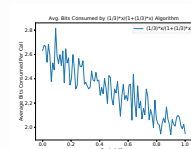
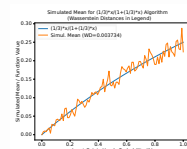
$$(1-x)*\tan(x)$$



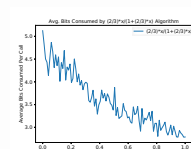
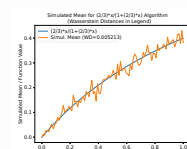
$$(1-x)/\cos(x)$$



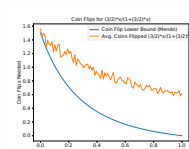
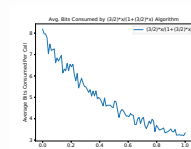
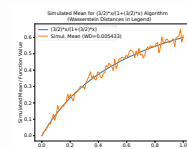
$$(1/3)*x/(1+(1/3)*x)$$



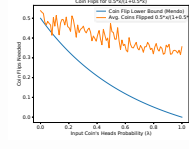
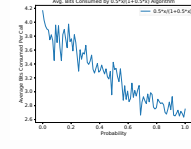
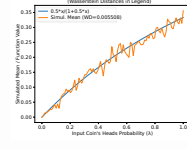
$$(2/3)*x/(1+(2/3)*x)$$



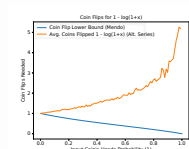
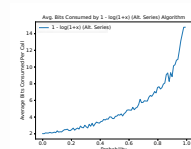
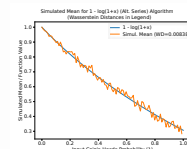
$$(3/2)*x/(1+(3/2)*x)$$



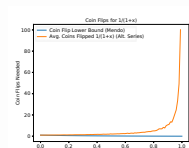
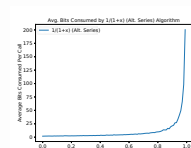
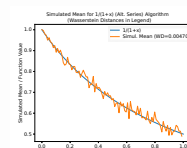
$$0.5*x/(1+0.5*x)$$



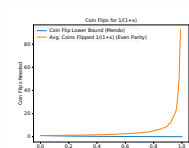
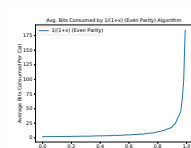
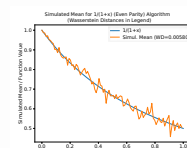
$$1 - \ln(1+x) \text{ (Alt. Series)}$$



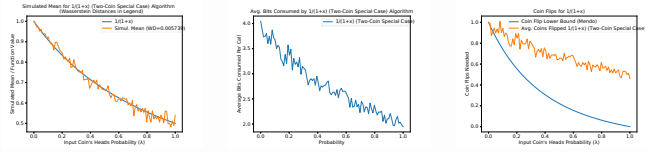
$$1/(1+x) \text{ (Alt. Series)}$$



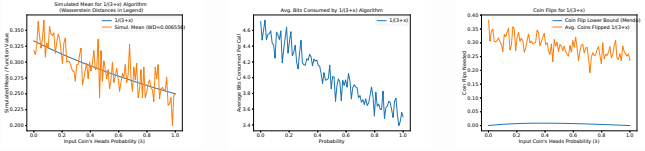
$$1/(1+x) \text{ (Even Parity)}$$



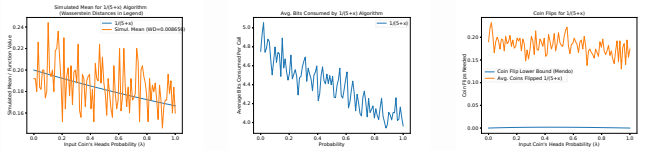
1/(1+x) (Two-Coin Special Case)



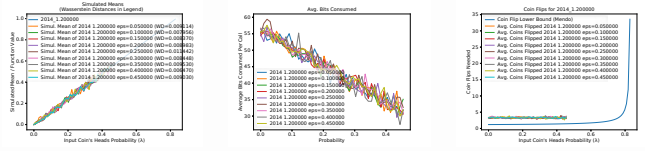
1/(3+x)



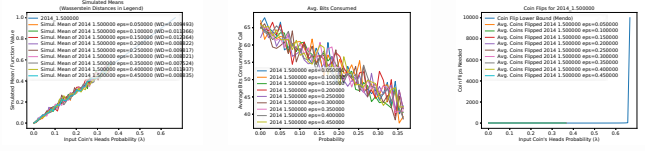
1/(5+x)



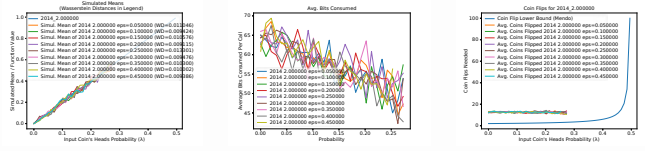
2014 1.200000
eps=0.050000



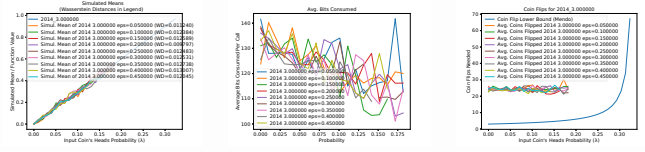
2014 1.500000
eps=0.050000



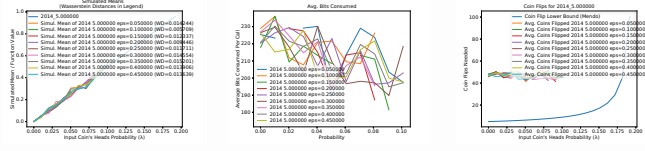
2014 2.000000
eps=0.050000



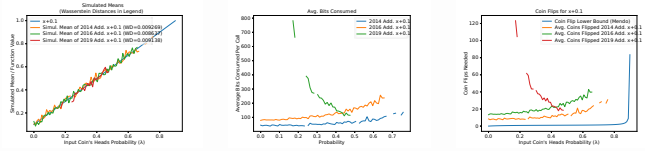
2014 3.000000
eps=0.050000



2014 5.000000
eps=0.050000



2014 Add. x+0.1



[illegible][illegible][illegible][illegible]

Figure 1 consists of three subplots. Subplot (a) is a scatter plot titled 'Scatter Plot Predicted vs. Actual Mean (Covariate)'. The x-axis is 'Input Covariate Mean Probability (C)' ranging from 0.0 to 0.5. The y-axis is 'Predicted Mean (Covariate)' ranging from 0.0 to 1.0. Data points are colored by year: 2014 (blue), 2015 (orange), 2016 (green), 2017 (red), 2018 (purple), 2019 (brown), 2020 (pink), and 2021 (grey). A diagonal line represents perfect prediction. Subplot (b) is titled 'Avg. Likelihood Ratio (ALR) vs. Probability'. The x-axis is 'Probability' from 0.0 to 0.5. The y-axis is 'Average Likelihood Ratio (ALR)' from 0.0 to 1.0. It shows curves for 2014-2021 and a 2014-2021 Mean. Subplot (c) is titled 'Cost Ratio vs. Input Covariate Mean Probability (C)'. The x-axis is 'Input Covariate Mean Probability (C)' from 0.0 to 0.5. The y-axis is 'Cost Ratio' from 0 to 100. It shows curves for 2014-2021 and a 2014-2021 Mean.

Figure 1 consists of three subplots labeled (a), (b), and (c), each showing a Receiver Operating Characteristic (ROC) curve. The x-axis for all plots is 'Input Coir's Weave Probability (x)' ranging from 0.00 to 0.10. The y-axis is 'Detection Rate (True Positive Rate)' ranging from 0.00 to 1.00.

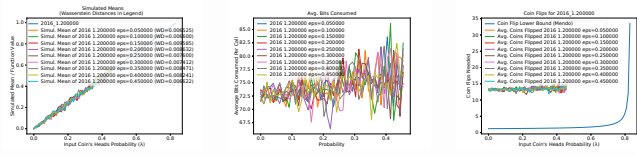
Subplot (a) is titled 'Proposed Model' and shows a single blue curve that starts at (0,0) and rises to (0.1, 1.0). A legend on the right lists the following parameters: $\alpha = 0.05$, $\beta = 0.05$, $\gamma = 0.05$, $\delta = 0.05$, $\epsilon = 0.05$, $\zeta = 0.05$, $\eta = 0.05$, $\theta = 0.05$, $\iota = 0.05$, $\kappa = 0.05$, $\lambda = 0.05$, $\mu = 0.05$, $\nu = 0.05$, $\xi = 0.05$, $\omicron = 0.05$, $\pi = 0.05$, $\rho = 0.05$, $\sigma = 0.05$, $\tau = 0.05$, $\upsilon = 0.05$, $\phi = 0.05$, $\chi = 0.05$, $\psi = 0.05$, $\omega = 0.05$, $\kappa = 0.05$, $\lambda = 0.05$, $\mu = 0.05$, $\nu = 0.05$, $\xi = 0.05$, $\omicron = 0.05$, $\pi = 0.05$, $\rho = 0.05$, $\sigma = 0.05$, $\tau = 0.05$, $\upsilon = 0.05$, $\phi = 0.05$, $\chi = 0.05$, $\psi = 0.05$, $\omega = 0.05$.

Subplot (b) is titled 'Proposed Model' and shows a single blue curve that starts at (0,0) and rises to (0.1, 1.0). A legend on the right lists the following parameters: $\alpha = 0.05$, $\beta = 0.05$, $\gamma = 0.05$, $\delta = 0.05$, $\epsilon = 0.05$, $\zeta = 0.05$, $\eta = 0.05$, $\theta = 0.05$, $\iota = 0.05$, $\kappa = 0.05$, $\lambda = 0.05$, $\mu = 0.05$, $\nu = 0.05$, $\xi = 0.05$, $\omicron = 0.05$, $\pi = 0.05$, $\rho = 0.05$, $\sigma = 0.05$, $\tau = 0.05$, $\upsilon = 0.05$, $\phi = 0.05$, $\chi = 0.05$, $\psi = 0.05$, $\omega = 0.05$, $\kappa = 0.05$, $\lambda = 0.05$, $\mu = 0.05$, $\nu = 0.05$, $\xi = 0.05$, $\omicron = 0.05$, $\pi = 0.05$, $\rho = 0.05$, $\sigma = 0.05$, $\tau = 0.05$, $\upsilon = 0.05$, $\phi = 0.05$, $\chi = 0.05$, $\psi = 0.05$, $\omega = 0.05$.

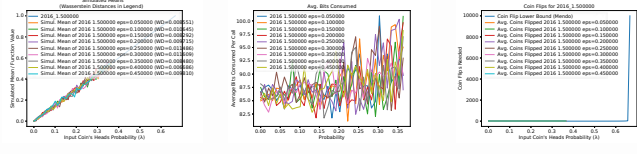
Subplot (c) is titled 'Proposed Model' and shows a single blue curve that starts at (0,0) and rises to (0.1, 1.0). A legend on the right lists the following parameters: $\alpha = 0.05$, $\beta = 0.05$, $\gamma = 0.05$, $\delta = 0.05$, $\epsilon = 0.05$, $\zeta = 0.05$, $\eta = 0.05$, $\theta = 0.05$, $\iota = 0.05$, $\kappa = 0.05$, $\lambda = 0.05$, $\mu = 0.05$, $\nu = 0.05$, $\xi = 0.05$, $\omicron = 0.05$, $\pi = 0.05$, $\rho = 0.05$, $\sigma = 0.05$, $\tau = 0.05$, $\upsilon = 0.05$, $\phi = 0.05$, $\chi = 0.05$, $\psi = 0.05$, $\omega = 0.05$, $\kappa = 0.05$, $\lambda = 0.05$, $\mu = 0.05$, $\nu = 0.05$, $\xi = 0.05$, $\omicron = 0.05$, $\pi = 0.05$, $\rho = 0.05$, $\sigma = 0.05$, $\tau = 0.05$, $\upsilon = 0.05$, $\phi = 0.05$, $\chi = 0.05$, $\psi = 0.05$, $\omega = 0.05$.

[illegible][illegible]

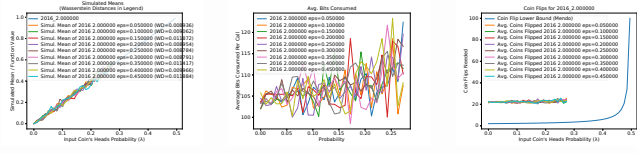
```
2016 1.200000
eps=0.050000
```



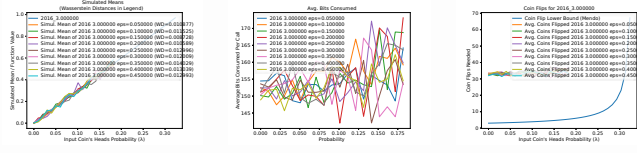
```
2016 1.500000
eps=0.050000
```



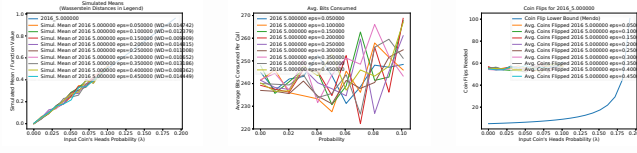
```
2016 2.000000
eps=0.050000
```



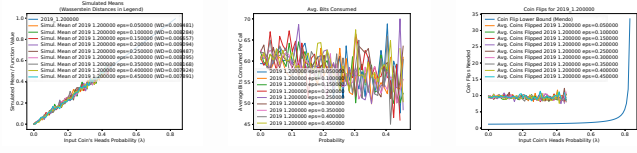
```
2016 3.000000
eps=0.050000
```



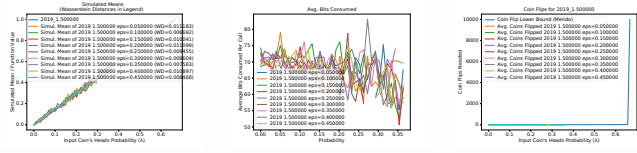
```
2016 5.000000
eps=0.050000
```



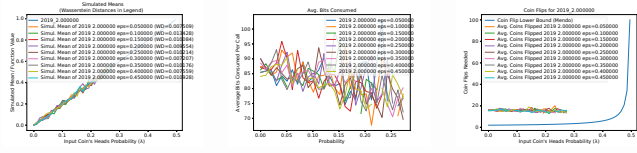
```
2019 1.200000
eps=0.050000
```



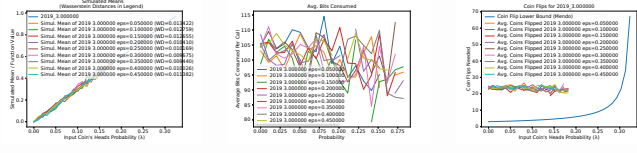
```
2019 1.500000
eps=0.050000
```



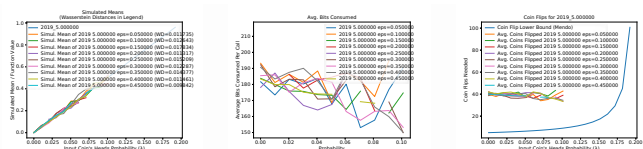
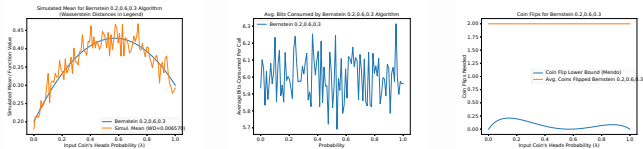
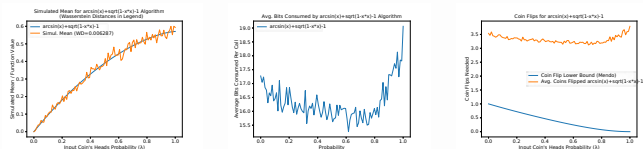
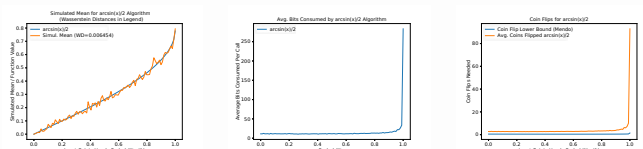
```
2019 2.000000
eps=0.050000
```



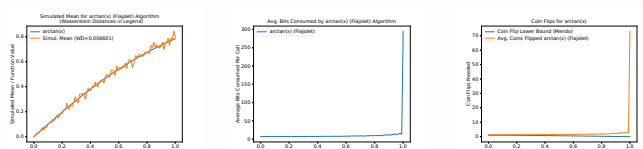
2019 3.000000
eps=0.050000



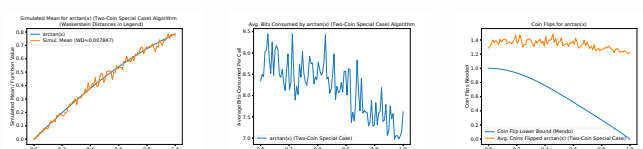
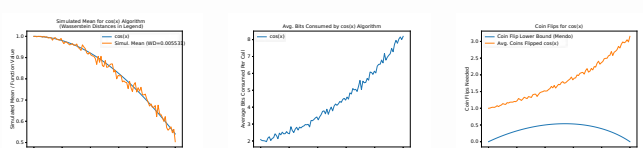
```
2019 5.000000
eps=0.050000
```

Bernstein
0.2,0.6,0.3
$$\arcsin(x) + \sqrt{1-x^2} - 1$$
 $\arcsin(x)/2$ 

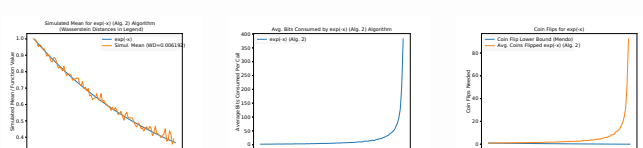
arctan(x)
(Flajolet)



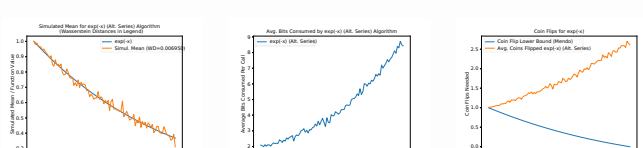
arctan(x) (Two-Coin Special Case)

 $\cos(x)$ 

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



exp(-x) (Alt.
Series)



Estimated Error for exact vs. (Proposed) Algorithm
 Scaled Error (log scale) vs. Scaled Inverse Correlation Length

Avg. Bits Complexity for exact vs. (Proposed) Algorithm
 Average Bit Complexity vs. Scaled Inverse Correlation Length

Cost (Flops) for exact vs. (Proposed) Algorithm
 Cost (Flops) vs. Scaled Inverse Correlation Length

Figure 1 consists of three subplots labeled (a), (b), and (c), each showing performance metrics against the input data's smooth probability (x-axis, ranging from 0.0 to 1.0).

- (a) Scalable Mean for expected T_1 :** The y-axis is 'Scalable Mean for expected T_1 ' (ranging from 0.0 to 1.4). The legend includes 'expected T_1 ' (blue line), 'observed T_1 ' (orange line), and 'observed T_1 - Bias (0.00718)' (green line). The observed T_1 starts at approximately 1.2 and decreases to about 0.4. The observed T_1 minus bias starts at approximately 1.1 and decreases to about 0.3.
- (b) Avg. Bias Computed by expected T_1 Algorithm:** The y-axis is 'Avg. Bias Computed by expected T_1 Algorithm' (ranging from 0 to 1.5). The legend includes 'expected T_1 ' (blue line). The bias starts at approximately 0.1 and increases to about 1.4.
- (c) Avg. RMSE Computed by expected T_1 Algorithm:** The y-axis is 'Avg. RMSE Computed by expected T_1 Algorithm' (ranging from 0.0 to 2.5). The legend includes 'Cox-Fit Linear Band Bound' (blue line), 'Cox-Fit Linear Band Bound' (orange line), and 'Avg. RMSE expected T_1 ' (green line). The Cox-Fit Linear Band Bound starts at approximately 0.8 and increases to about 2.4. The Avg. RMSE expected T_1 starts at approximately 0.8 and increases to about 2.4.

Figure 1 consists of three subplots showing the performance of the proposed algorithm for a sigmoidal target.

- Left Plot:** Standard Mean Squared Error (MSE) vs. $\log_2(x)$. The y-axis ranges from 0.0 to 0.2. The x-axis ranges from 0 to 14. The legend indicates:
 - $\log_2(x)$ (blue line with circles)
 - $\log_2(x)$ (Proposed) (orange line with circles)
 - Linear Mean (MSE=0.088215) (red line)
 The proposed algorithm's MSE is significantly lower than the linear mean, closely following the $\log_2(x)$ curve.
- Middle Plot:** Avg. Bits Consumed by $\log_2(x)$ (Proposed) Algorithm vs. $\log_2(x)$. The y-axis ranges from 0 to 10. The x-axis ranges from 0 to 14. The legend indicates:
 - $\log_2(x)$ (Proposed) (blue line)
 The bits consumed increase linearly with $\log_2(x)$ until approximately $\log_2(x) = 12$, after which they increase sharply.
- Right Plot:** Count Rate for $\log_2(x)$ vs. $\log_2(x)$. The y-axis ranges from 0 to 14. The x-axis ranges from 0 to 14. The legend indicates:
 - Count Rate Linear Bound (Shaded) (blue shaded area)
 - Count Rate Upper Bound (Proposed) (orange line)
 - Count Rate Lower Bound (Proposed) (blue line)
 The proposed algorithm's count rate is bounded by the linear bound and the upper bound, showing a sharp increase after $\log_2(x) = 12$.

Figure 1 consists of three subplots comparing the proposed algorithm with the state-of-the-art. The left subplot shows the Graded Mean Error (log10) versus log10(1/epsilon) for two values of log10(1/epsilon): 0.000001 (blue line) and 0.0000001 (orange line). The middle subplot shows the Average Work Computed by log10(1/epsilon) versus log10(1/epsilon) for the same two values. The right subplot shows the Cost (log) versus log10(1/epsilon) for the same two values. The x-axis for all plots is log10(1/epsilon) ranging from -0.6 to 1.6. The y-axis for the left plot is Graded Mean Error (log10) ranging from -0.4 to 0.4. The y-axis for the middle plot is Average Work Computed by log10(1/epsilon) ranging from 0.0 to 1.0. The y-axis for the right plot is Cost (log) ranging from 0.0 to 1.8.

Figure 1 consists of four subplots arranged in a 2x2 grid, illustrating the performance of the proposed algorithm. The top-left plot shows 'Simulated Results for points 1,2,3 Algorithm' with 'Distorted Noise (dB)' on the y-axis (0.0 to 1.8) and 'Signal-to-Noise Ratio (dB)' on the x-axis (0.0 to 1.6). It compares 'points 1,2,3' (blue line) and 'Linear Search (0.0-0.000000)' (red line). The top-right plot shows 'Avg. Bits Consumed by points 1,2,3 Algorithm' with 'Average Bits Consumed (Kbit)' on the y-axis (0 to 4000) and 'Signal-to-Noise Ratio (dB)' on the x-axis (0.0 to 1.6). The bottom-left plot shows 'Avg. Bits Consumed by points 1,2,3 Algorithm' with 'Average Bits Consumed (Kbit)' on the y-axis (0 to 4000) and 'Signal-to-Noise Ratio (dB)' on the x-axis (0.0 to 1.6). The bottom-right plot shows 'Cost Files for points 1,2,3' with 'Cost (Kbit)' on the y-axis (0 to 400) and 'Signal-to-Noise Ratio (dB)' on the x-axis (0.0 to 1.6). It compares 'Cost File Linear Search Method' (blue line), 'Cost File Exhaustive Search Method' (red line), and 'Avg. Cost Proposed points 1,2,3' (green line).

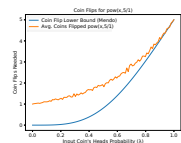
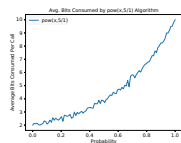
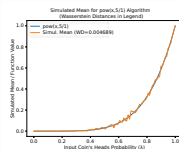
Figure 1 consists of three subplots labeled (a), (b), and (c), comparing the performance of the proposed algorithm (green line) with the state-of-the-art algorithm (blue line).

- (a) Standard Error vs. Signal-to-Noise Ratio:** The x-axis is 'Signal-to-Noise Ratio (dB)' ranging from 0.0 to 1.0. The y-axis is 'Standard Error (Log10)' ranging from 0.0 to 1.5. The legend indicates 'proposed' (green line) and 'State-of-the-art (2016) [28]' (blue line). Both lines show an increasing trend, with the proposed algorithm's error being slightly lower than the state-of-the-art's error across the range.
- (b) Average Bit Consumption vs. Probability:** The x-axis is 'Probability' ranging from 0.0 to 1.0. The y-axis is 'Average Bit Consumption (bits)' ranging from -2.0 to 6.0. The legend indicates 'proposed' (green line) and 'proposed' (blue line). The proposed algorithm's bit consumption is consistently lower than the state-of-the-art's bit consumption across the range.
- (c) Cost Flips vs. Probability:** The x-axis is 'Signal-to-Noise Ratio (dB)' ranging from 0.0 to 1.0. The y-axis is 'Cost Flips' ranging from 0.000 to 2.000. The legend indicates 'Cost Flip Lower Bound Proposed' (green line) and 'Avg. Cost Flips proposed' (blue line). The proposed algorithm's cost flips are consistently lower than the state-of-the-art's cost flips across the range.

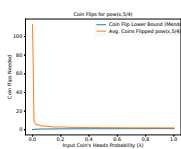
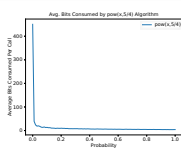
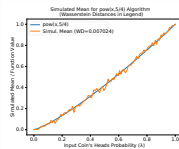
Figure 1 consists of three subplots comparing the proposed A2C algorithm with the state-of-the-art algorithms.

- Left Plot:** Standard Error of Gradient vs. Total Sample Complexity (N). The y-axis ranges from 0.0 to 1.0, and the x-axis ranges from 0.0 to 1.0. The legend indicates "proposed A2C" (blue line) and "State-of-the-art (MSE=0.036882)" (orange line). Both lines show a decreasing trend, with the proposed A2C line generally lower than the state-of-the-art line.
- Middle Plot:** Avg. Bias Observed by power-A2C Algorithm vs. Sample Complexity (N). The y-axis ranges from 0.0 to 0.4, and the x-axis ranges from 0.0 to 1.0. The legend indicates "power-A2C" (blue line). The bias starts high (around 0.35) and decreases rapidly, approaching zero as sample complexity increases.
- Right Plot:** Cost Files for power-A2C vs. Power Coeff. (Power-Optimal). The y-axis ranges from 0.0 to 4.0, and the x-axis ranges from 0.0 to 1.0. The legend indicates "Avg. Cost Files power-A2C" (blue line) and "Avg. Cost Files power-A2C" (orange line). Both lines show a decreasing trend, with the proposed A2C line generally lower than the state-of-the-art line.

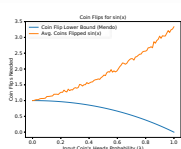
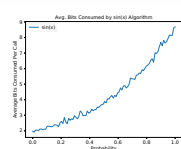
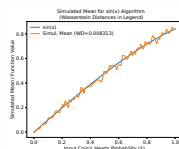
pow(x,5/1)



pow(x,5/4)



sin(x)



sqrt(x)

