

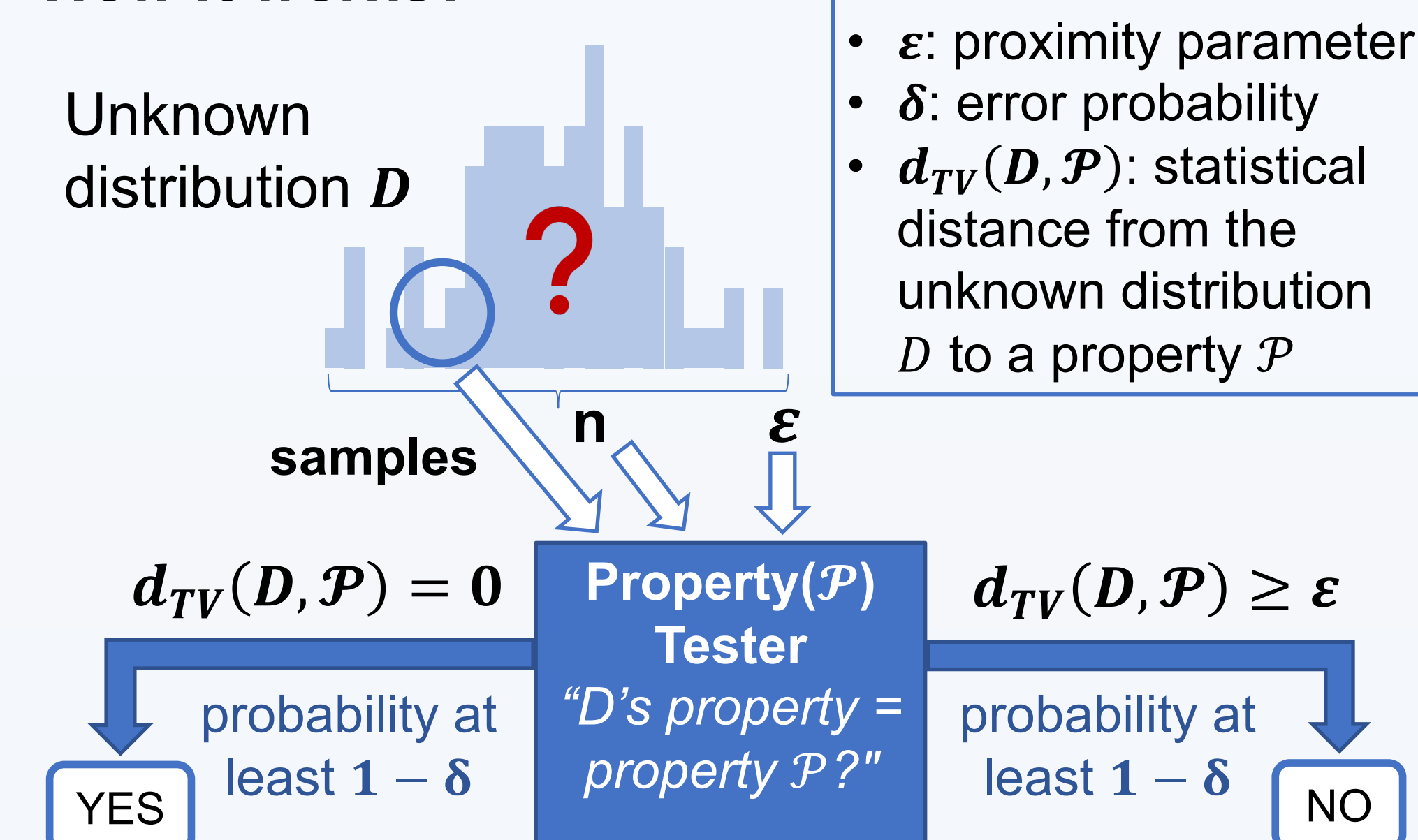
## MOTIVATION

- Sample-efficient distribution testing algorithms have yet to be implemented and utilized
- Specific number of samples required for these algorithms remain undetermined
- Theoretical claims of papers need to be verified

## BACKGROUND

**Distribution Testing** Test if an unknown distribution has some property

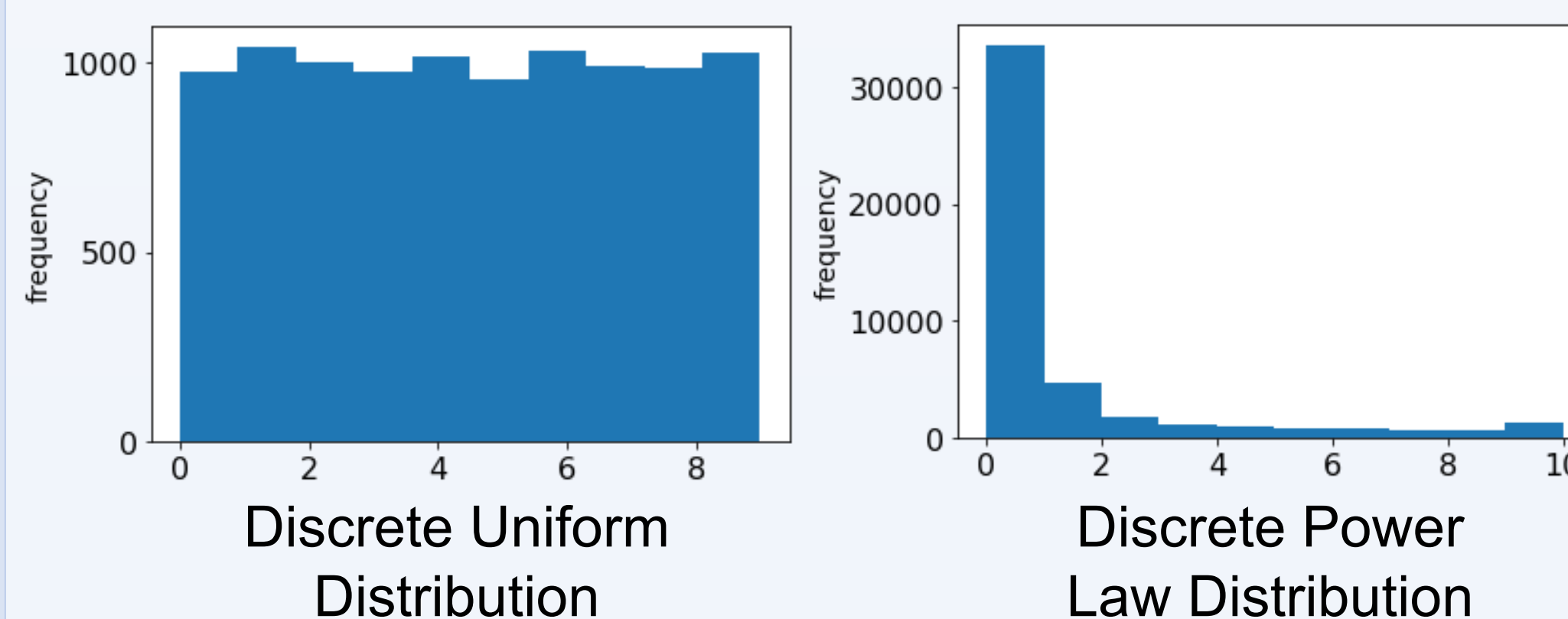
**How it works?**



**Uniformity Testing** Test if all discrete bins have the same weight

**Identity Testing** Test whether an unknown distribution matches a known distribution

EX. Is distribution  $p$  a power law distribution?



**Potential Applications of Distribution Testing**

- Check if the distribution of NY State Lotto's winning numbers is uniform
- Determine if the dataset to regress is linear by checking if residual is normally distributed

## PROBLEM STATEMENT

- To provide working implementations of theoretically-efficient distribution testing algorithms, with determined sample-complexity and relative run-times

## METHODS

**Implemented Algorithms**

| Uniformity tester         | Identity tester         |
|---------------------------|-------------------------|
| $l_2$ estimator           | Chi-squared tester      |
| Coincidence-based tester  | instance-optimal tester |
| "New approach" uniformity |                         |

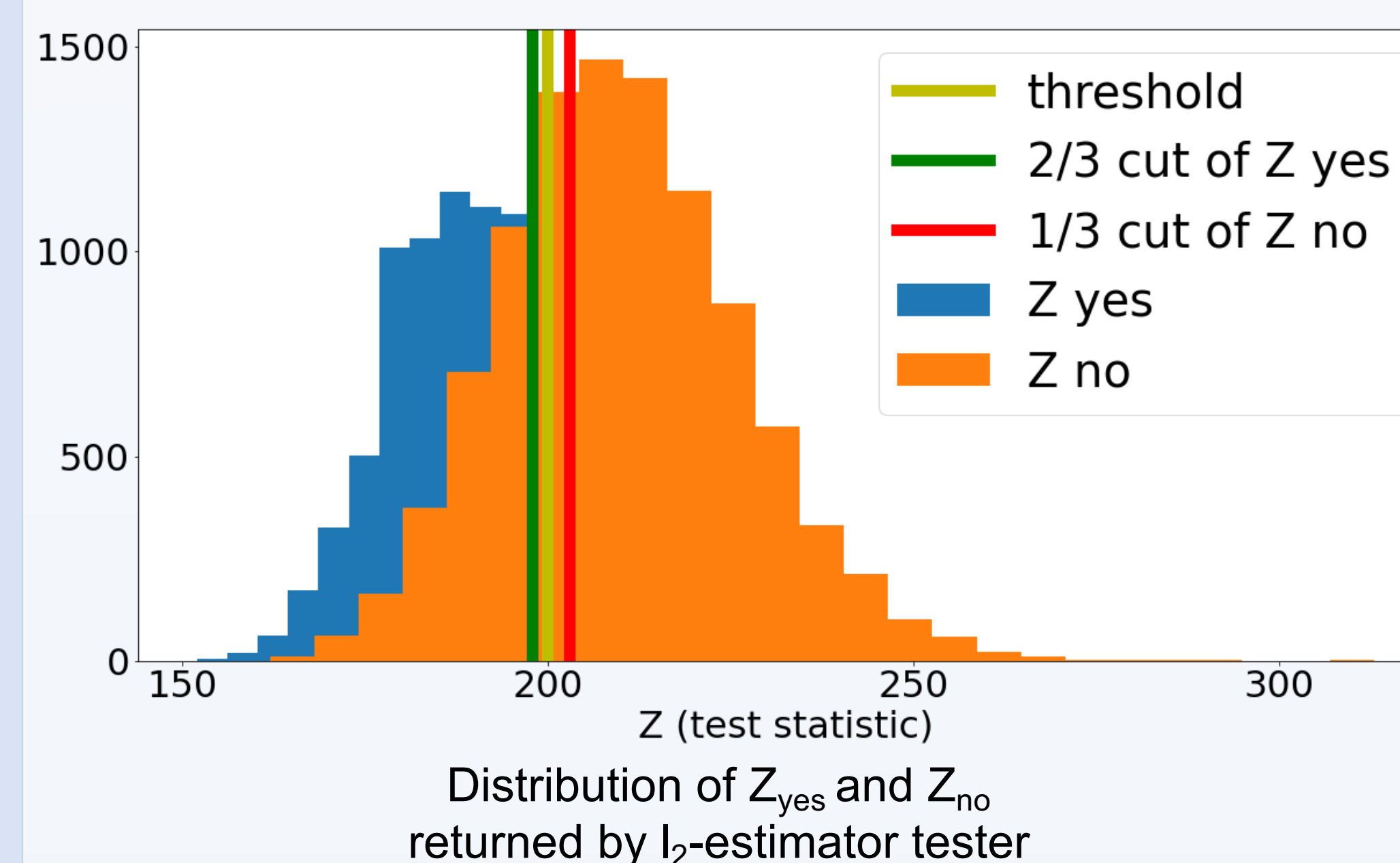
**How to Pin Down Constants?**

Algorithms compute test statistic  $Z$  on tested distribution and use  $Z$  to indicate if the distribution has the desired property.

**STEP 1** Run the algorithms on both yes and no distributions for  $10^6$  times, to get  $Z_{\text{yes}}$  and  $Z_{\text{no}}$ .

**STEP 2** Adjust number of samples to make the  $Z_{\text{yes}}$  and  $Z_{\text{no}}$  distribution overlap at percentile of  $100 * \delta$  (ex.  $\delta = \frac{1}{3}$ , intersect at 33<sup>rd</sup> percentile).

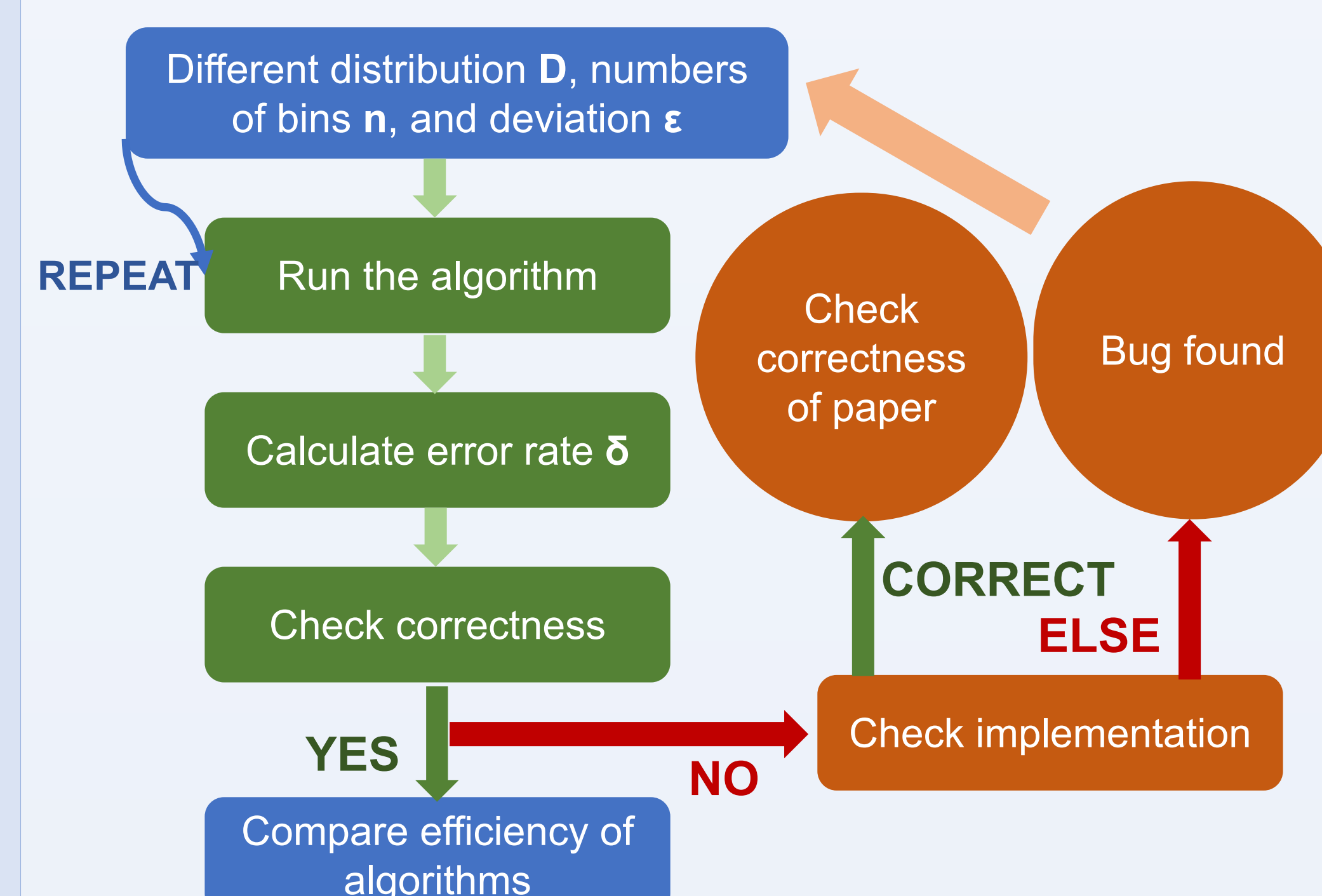
**STEP 3** Adjust the constant of threshold, making the threshold match the "true" threshold under all circumstances.



**Testing Distributions**

|                             |                                       |
|-----------------------------|---------------------------------------|
| Uniformity Yes Distribution | Uniform D. over $N$                   |
| Uniformity No Distribution  | $(1 \pm \epsilon)/n$ for all $n$ bins |
| Identity Yes Distribution   | $q_i = c_1(i+c_2)^{-3/2}$             |
| Identity No Distribution    | $p_i = q_i(1 \pm \epsilon)$           |

**How to make sure constants are correct?**



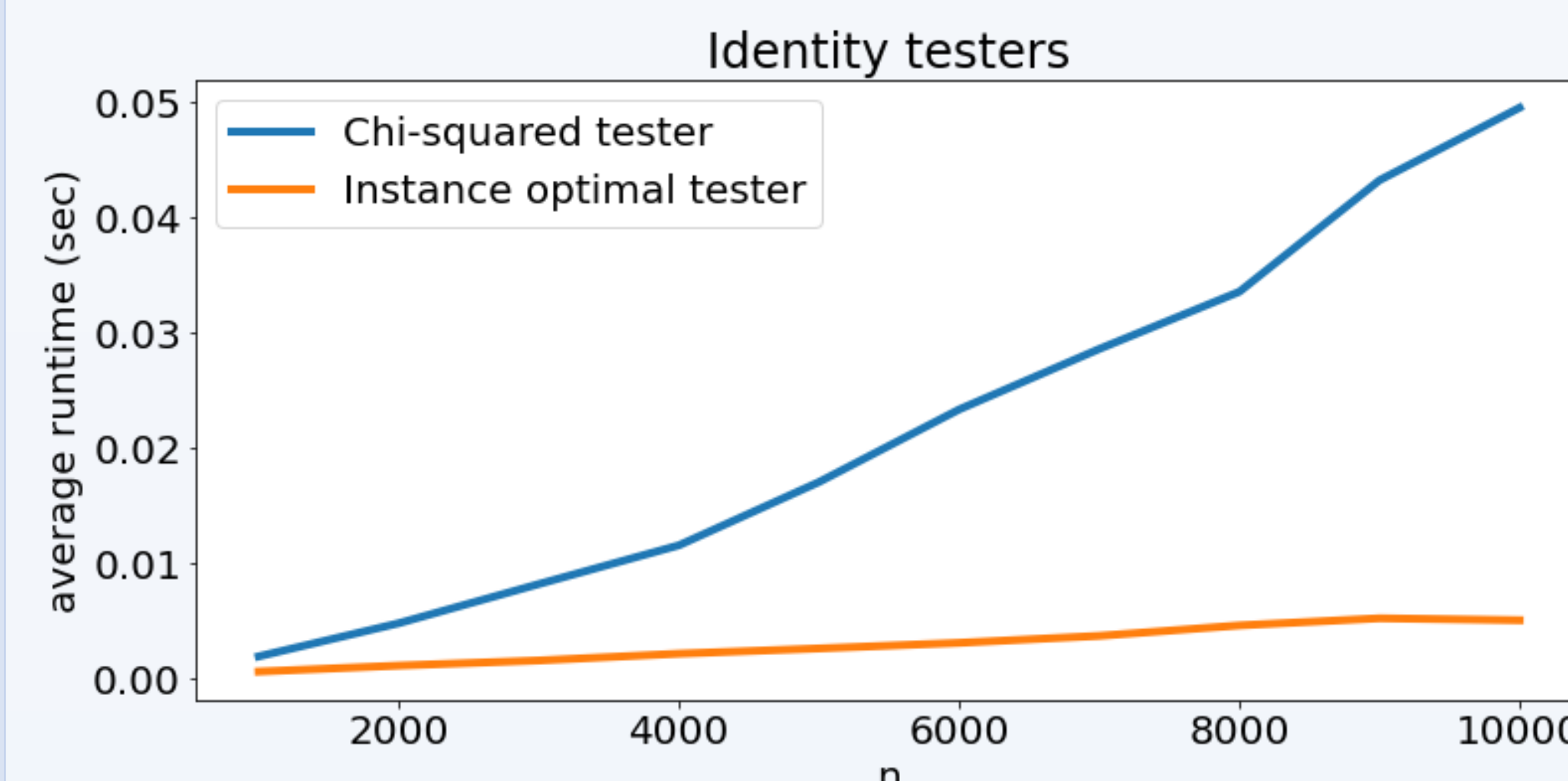
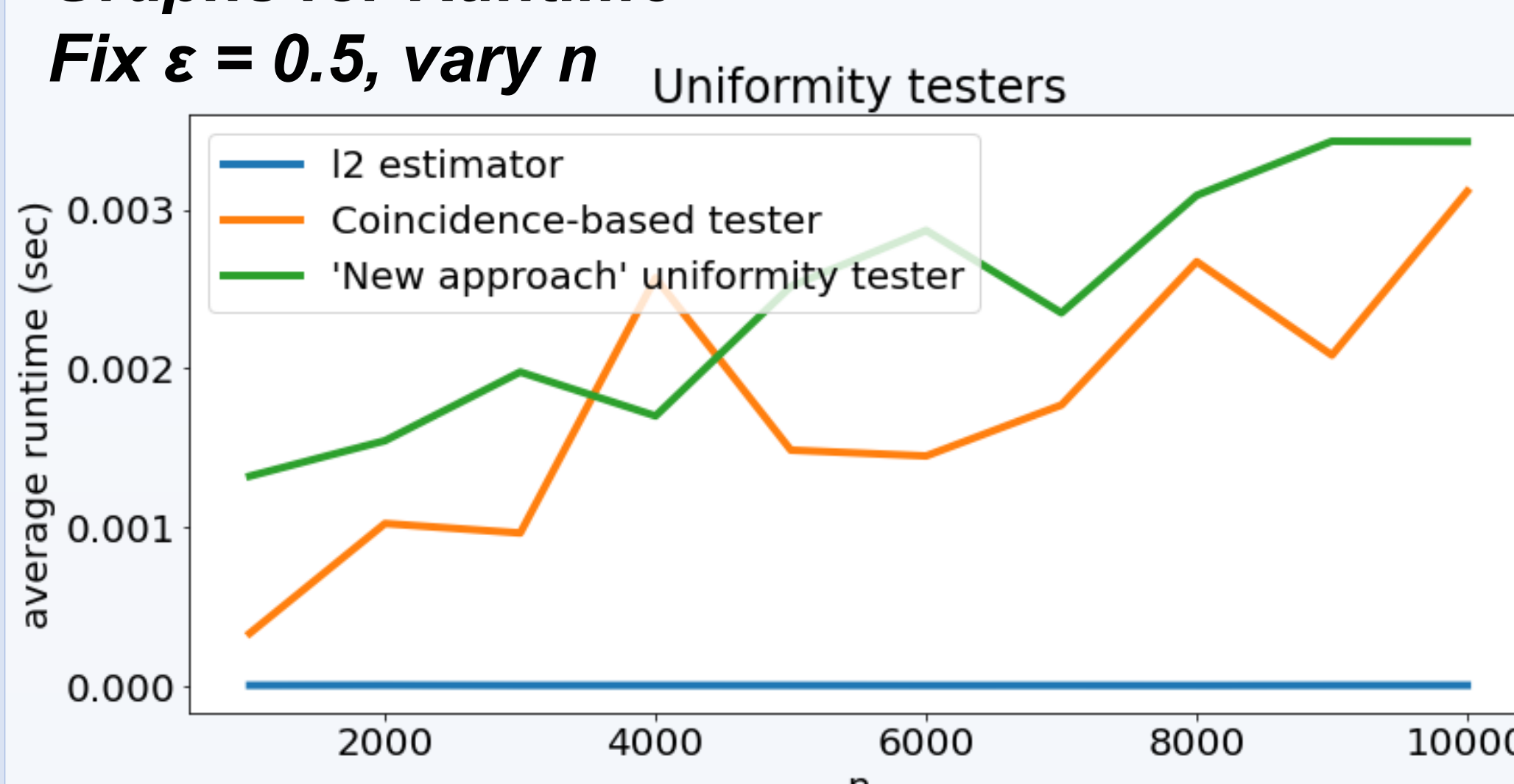
## RESULTS & COMPARISONS

**Worst-case Sample Size Pinned Down**

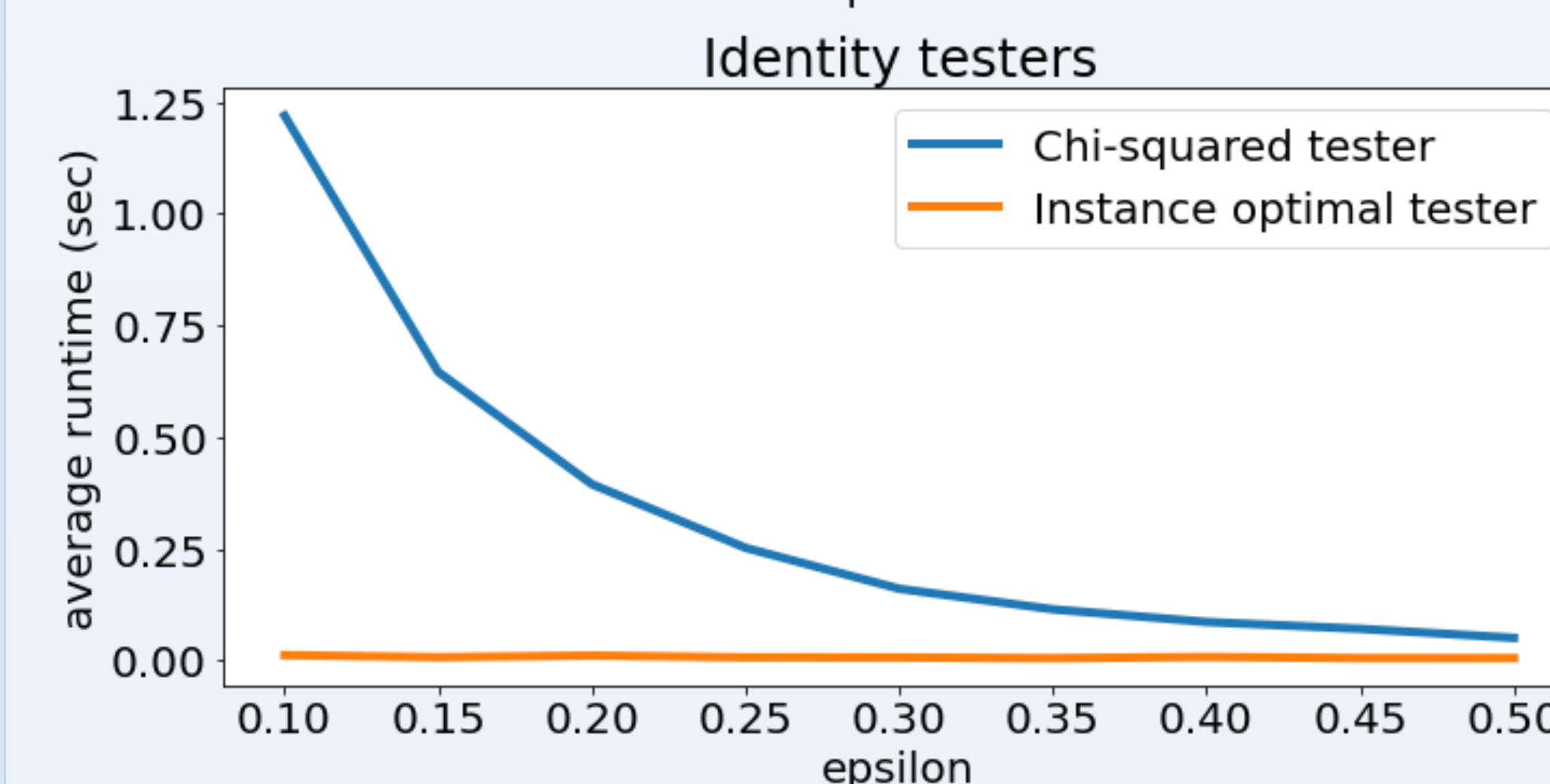
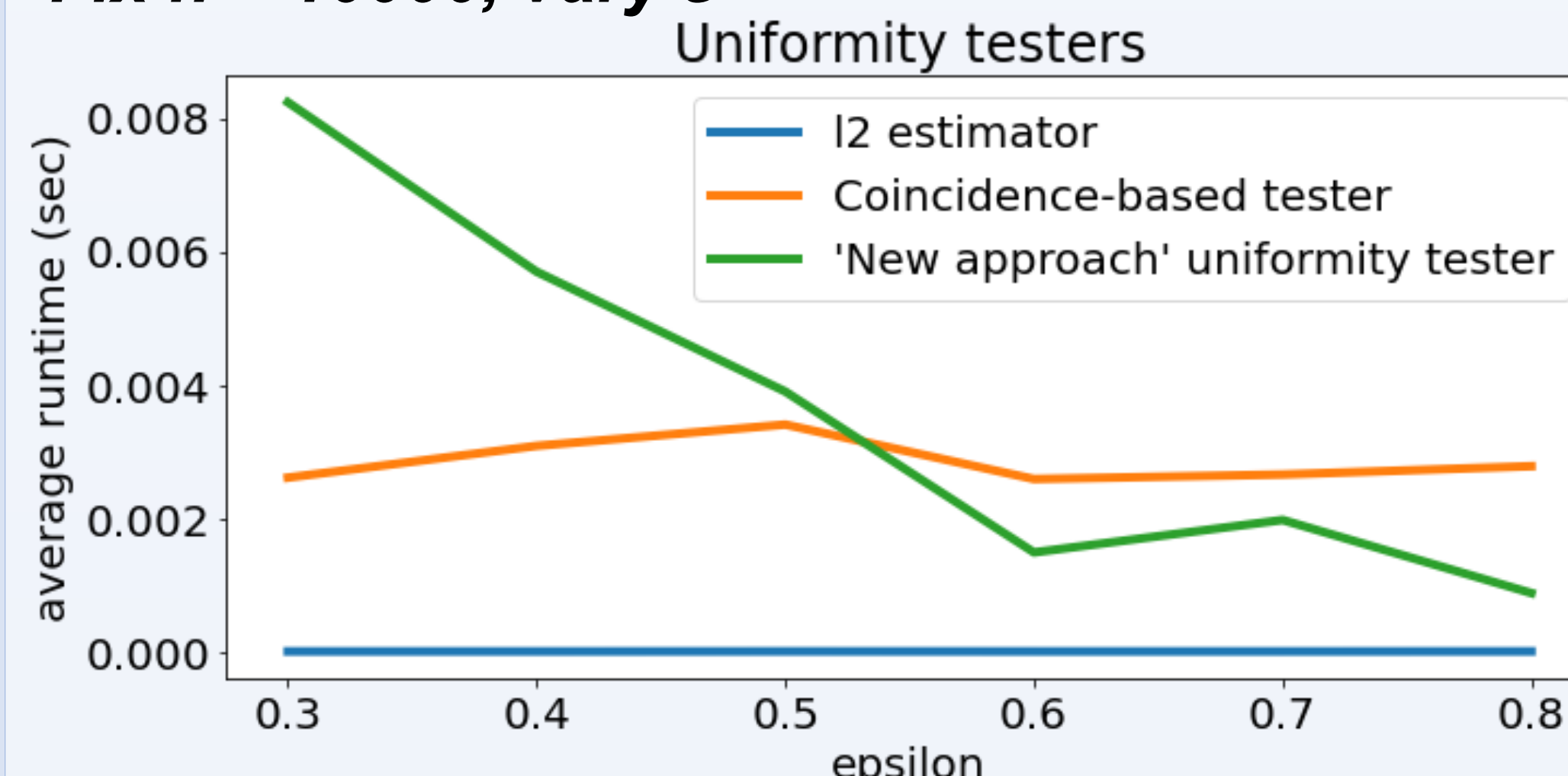
| Uniformity tester                | Sample size                  |
|----------------------------------|------------------------------|
| $l_2$ estimator                  | $1.76 * \sqrt{n}/\epsilon^2$ |
| Coincidence-based tester         | $2.1 * \sqrt{n}/\epsilon^2$  |
| "New approach" uniformity tester | $5.9 * \sqrt{n}/\epsilon^2$  |

| Identity tester         | Sample size  |
|-------------------------|--|
| Chi-squared tester      | $1.45 * \sqrt{n}/\epsilon^2$                                     |
| Instance optimal tester | $1.8 * \max(\frac{1}{\epsilon}, \ p\ _{\frac{1}{2}}/\epsilon^2)$ |

**Graphs for Runtime**



**Fix  $n = 10000$ , vary  $\epsilon$**



**Sample Complexity Analysis**

- The worst-case sample size required for each tester is proportional to  $\sqrt{n}/\epsilon^2$  ( $1.45 \leq C \leq 5.9$  for  $C * \sqrt{n}/\epsilon^2$ )
- Our experimental results match the theoretical expectations.
- $l_2$  estimator is the most sample-efficient uniformity tester.
- Chi-squared tester is more sample-efficient when known distribution( $q$ ) is uniform; less efficient if  $q$  is a power law distribution.

**Runtime Analysis**

Ran on MacBook Air  
Processor: 1.8 GHz Dual-core Intel Core i5  
Memory: 8 GB 1600 MHz DDR3

- All algorithms are fast enough to run on even general-purpose computers (~1 second).
- Outcome matches theoretical expectations.

| Uniformity tester                |                                |
|----------------------------------|--------------------------------|
| $l_2$ estimator                  | Fastest                        |
| Coincidence-based tester         | Constant with all $\epsilon$   |
| "New approach" uniformity tester | Faster as $\epsilon$ increases |

| Identity tester         |                             |
|-------------------------|-----------------------------|
| Chi-squared tester      | Fast with larger $\epsilon$ |
| Instance optimal tester | Faster                      |

## CONCLUSION

GitHub Repo for Implementations:



## FUTURE WORK

- Implementation of closeness and independence testers
- Research paper to compile and share findings

## ACKNOWLEDGEMENTS

We would like to thank Professor Kane and Sihan Liu for guiding our research, Professor Alvarado, Professor ElSherief, and Vaidehi Gupta for running the ERSP program and giving us this opportunity