Adaptive Hashing for Model Counting

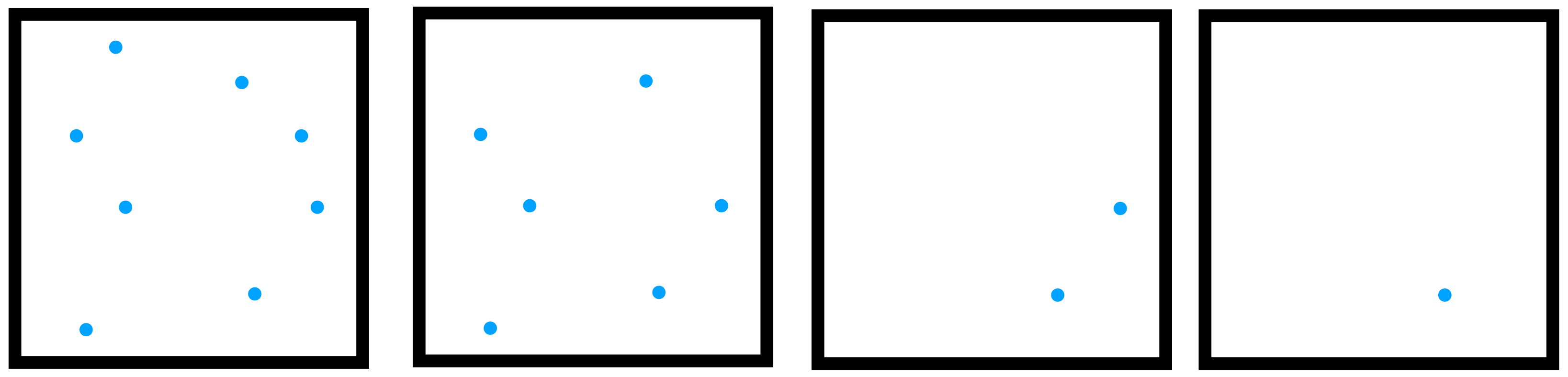
Jonathan Kuck, Tri Dao, Shengjia Zhao, Burak Bartan, Ashish Sabharwal, and Stefano Ermon

kuck@stanford.edu, trid@stanford.edu, sjzhao@stanford.edu, bbartan@stanford.edu, ashishs@allenai.org, ermon@stanford.edu

*TLDR: Hashing-based counting algorithms[1,2] currently operate identically for any shape of input instance. We **adapt** hash functions to specific instances using previously found solutions and use a novel **aggregation** strategy across independent repetitions to **improve statistical efficiency**. Our methods improve lower bounds by a median factor of 2^{13} over 1198 input instances.*

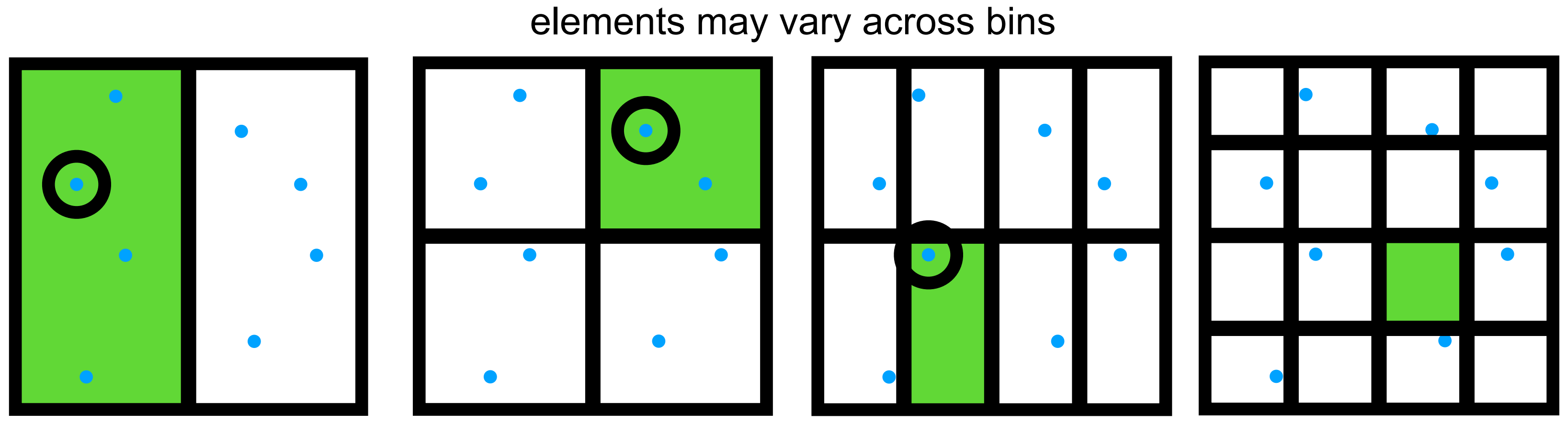
Background: Approximate Counting

The number of elements in a set can be estimated by randomly removing each element with probability 1/2. Repeating this operation k times, until only 1 element remains, gives 2^k as an estimate of the set's size. This approach is statistically optimal, but computationally intractable.



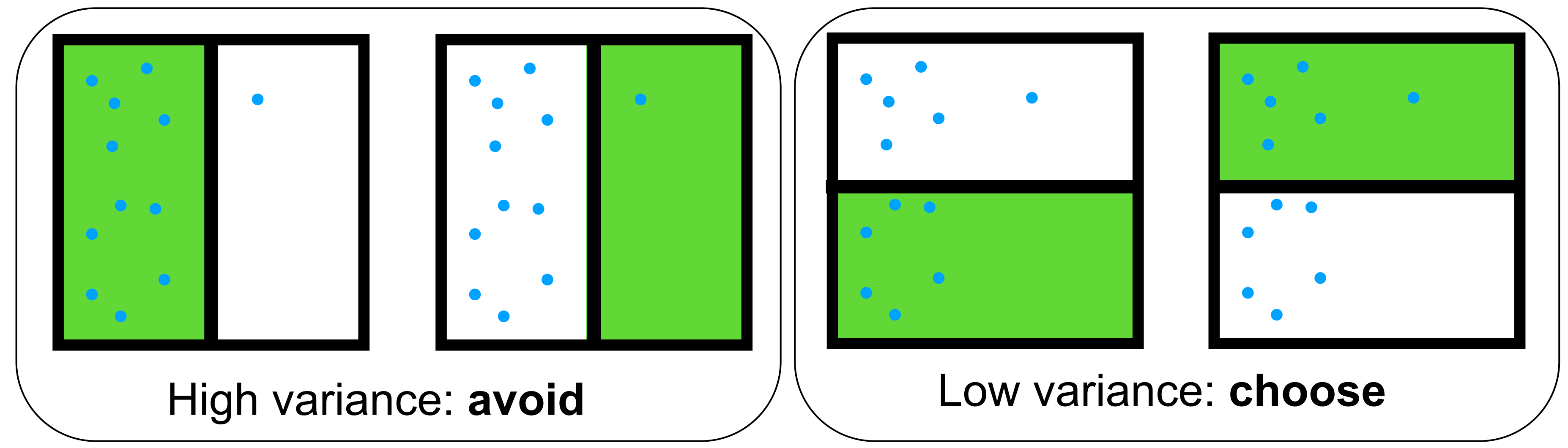
Background: Fixed Hashing

Hash functions provide a computationally efficient alternative. All elements in the set are hashed and a randomly selected bin is checked for elements. Repeating this process with increasingly more bins, until the random bin is empty, gives an estimate of the set size. Computational efficiency comes at the cost of low statistical efficiency as the number of elements may vary across bins



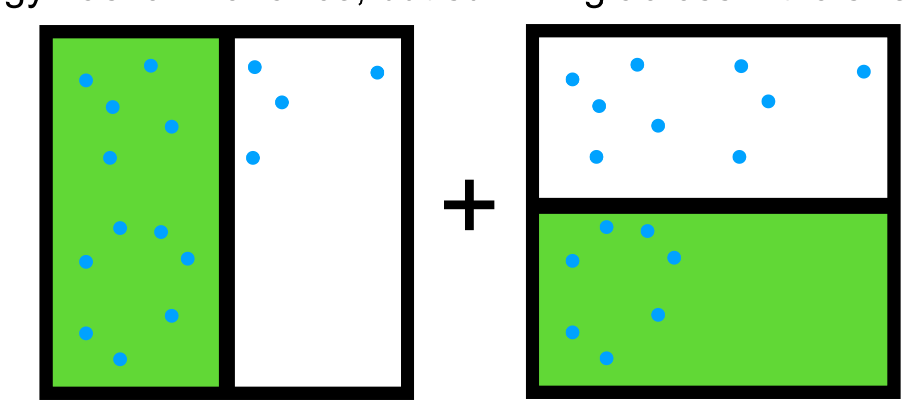
Adaptive Hashing

We choose low variance, cheap binning strategies for *specific instances* based on previously found solutions, resulting in improved statistical efficiency with high computational efficiency.



Aggregation Across Trials

We sample independent binning strategies and aggregate the solutions. This improves statistical efficiency, but requires additional parallel computation. Note that in this example neither binning strategy has low variance, but summing across 2 trials reduces the variance.



Algorithm

Algorithm 2 New, Adaptive Lower Bound

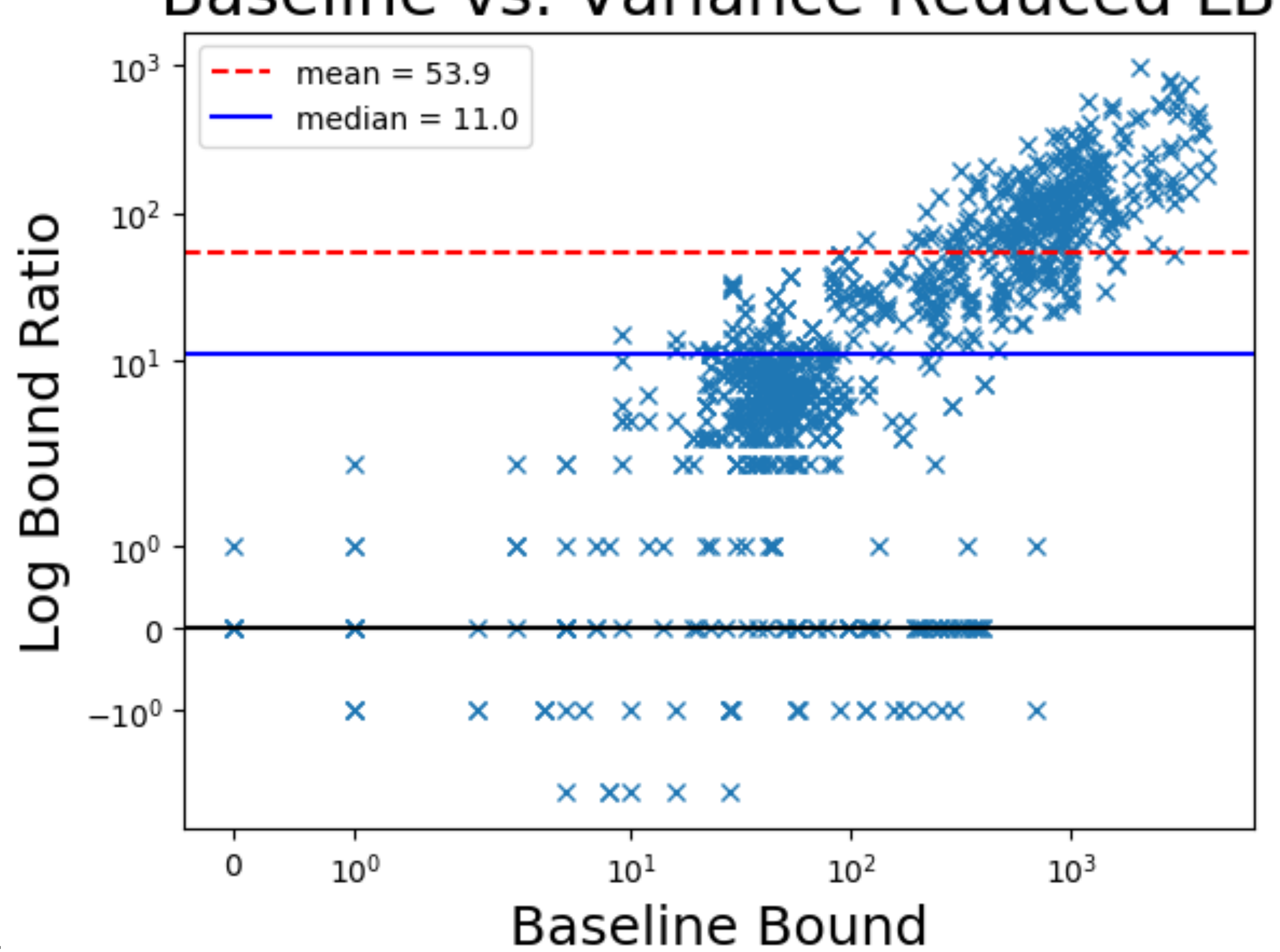
Inputs: s: Solution cutoff
Δ: Failure probability
K: Number of repetitions per trial
$\mathcal{O}_S$: A SAT oracle
Output: A probabilistic lower bound on $|S|$

1: $T = \lceil 8 \log \frac{1}{\Delta} \rceil$
2: $m = 1$
3: $\mathcal{D} = \emptyset$ // Solutions found so far
4: **while** $m \leq n$ **do**
5:   **for** $t = 1, \cdots, T$ **do**
6:     **for** $k = 1, \cdots, K$ **do**
7:       $A^m = \text{SampleAdaptiveMatrix}(\mathcal{D}, m)$
8:       Sample $b \sim \text{Uniform}(\mathbb{F}_2^m)$
      // Invoke oracle $\mathcal{O}_S$ up to sK times to check
      // whether the input formula with additional
      // constraints $A^m x = b$ has at least sK
      // distinct solutions
9:       $w_k = \min\left\{sK, |S \cap (h^m)^{-1}(b)|\right\}$
10:       $S' =$ Up to sK solutions obtained at step 9
11:       $\mathcal{D} = \mathcal{D} \cup S'$
12:     $w^t = \min\left\{s, \frac{1}{K}\sum_{k=1}^{K} w_k\right\}$
13:   **if** $\sum_{t=1}^{T} w^t < sT/2$ **then**
14:     break
15:   $m = m + 1$
16: Output "$|S| \geq s\lfloor 2^{m-3} \rfloor$"

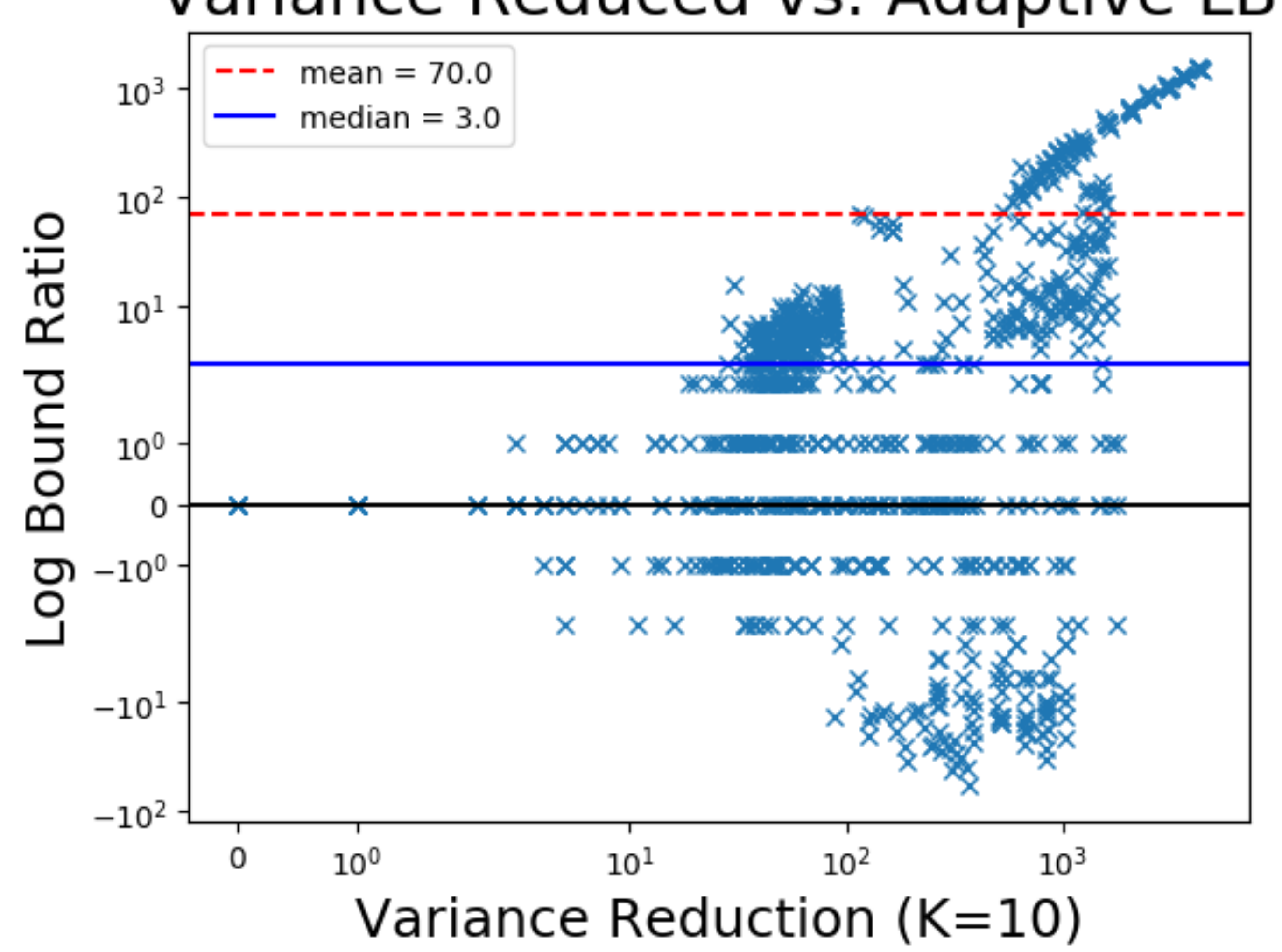
Empirical Study

Our baseline family of fixed hashed functions[1] was inspired by LDPC codes, the current state of the art in ultra-efficiency. Left: aggregation improves the lower bound. Middle: adaptation with aggregation improves the lower bound compared with aggregation alone. Right: tradeoff in statistical and computational efficiency for our method and the baseline on a single challenging instance. We can improve both the lower bound and runtime by orders of magnitude.

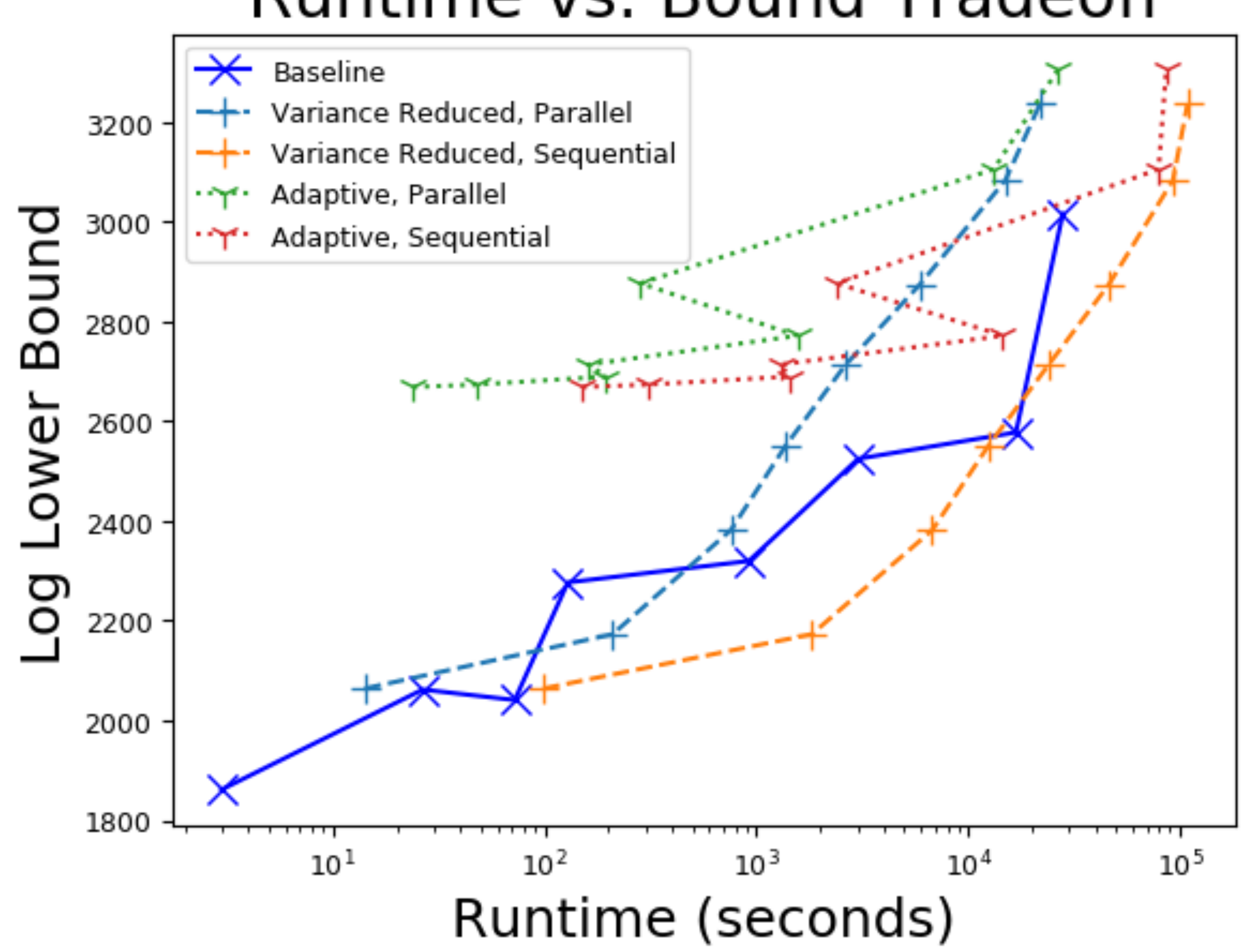
Baseline vs. Variance Reduced LB



Variance Reduced vs. Adaptive LB



Runtime vs. Bound Tradeoff



1. Achlioptas, Dimitris, Zayd Hammoudeh, and Panos Theodoropoulos. "Fast and Flexible Probabilistic Model Counting." *International Conference on Theory and Applications of Satisfiability Testing* 2018.
2. Chakraborty, Supratik, Kuldeep S. Meel, and Moshe Y. Vardi. "Algorithmic Improvements in Approximate Counting for Probabilistic Inference: From Linear to Logarithmic SAT Calls." *IJCAI* 2016.