New Algorithms and Applications for Risk-Limiting Audits

Bar Karov
Weizmann Institute of Science, Rehovot, Israel

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Abstract

Risk-limiting audits (RLAs) are a significant tool in increasing confidence in the accuracy of elections. They consist of randomized algorithms which check that an election's vote tally, as reported by a vote tabulation system, corresponds to the correct candidates winning. If an initial vote count leads to the wrong election winner, an RLA guarantees to identify the error with high probability over its own randomness. These audits operate by sequentially sampling and examining ballots until they can either confirm the reported winner or identify the true winner.

The first part of this work suggests a new generic method, called "Batchcomp", for converting classical (ballot-level) RLAs into ones that operate on batches. As a concrete application of the suggested method, we develop the first RLA for the Israeli Knesset elections, and convert it to one which operates on batches using "Batchcomp". We ran this suggested method on the real results of recent Knesset elections.

The second part of this work suggests a new use-case for RLAs: verifying that a population census leads to the correct allocation of parliament seats to a nation's federal-states. We present an adaptation of ALPHA [12], an existing RLA method, to a method which applies to censuses. This suggested census RLA relies on data from both the census and from an additional procedure which is already conducted in many countries today, called a post-enumeration survey.

2012 ACM Subject Classification Applied computing → Voting / election technologies

Keywords and phrases Risk-Limiting Audit, RLA, Batch-Level RLA, Census

Digital Object Identifier 10.4230/LIPIcs.FORC.2023.2

Supplementary Material *Software (Source Code and Additional Plots)*: `https://github.com/TGKar/Batch-and-Census-RLA`; archived at `swh:1:dir:9bd16e71658b0883e3ac966d48f81f48310fc9f3`

Funding Research supported in part by grants from the Israel Science Foundation (no.2686/20), by the Simons Foundation Collaboration on the Theory of Algorithmic Fairness and by the Israeli Council for Higher Education (CHE) via the Weizmann Data Science Research Center.
Moni Naor: Incumbent of the Judith Kleeman Professorial Chair.

1 Introduction

Running an election is a delicate endeavour, since casting and tallying votes entails seemingly contradictory requirements: counting the votes should be accurate and it must also be confidential. A risk-limiting audit (RLA) is a process whose goal is to increase the confidence that results of an election were tallied appropriately, or more accurately that the winner/s were chosen correctly. It is usually described for election systems where there is an electronic vote tabulation, whose tally is referred to as the **reported results**, but also backup paper-ballots, whose tally is assumed to be the **true results**. The procedure examines what is hopefully a relatively small number of the backup paper-ballots, and comparing them to the full reported results of the electronic voting system. These audits are randomized algorithms, where the randomization is manifested in the choice of ballots to examine, and potentially the order in which they are examined.

A risk-limiting audit ends either when the reported winners of the election are confirmed, or after a full recount of the backup paper-ballots of all voters. The audit's goal is to confirm that the reported winners according to the electronic vote tabulation (the reported tally) match the winners according to the paper-backups (the true tally). Note that RLAs verify that the elections resulted in the correct winners according to the backup paper-ballots, and not that the reported vote tally was completely accurate; an RLA will approve election results that contain counting errors which do not change the winners of the elections. This fact is useful since it would be infeasible to expect the vote tally to be accurate up to every single ballot, but we should avoid at all cost counting errors which change the winners of the elections.

The claimed guarantee of RLAs is that if the reported winners of the elections are not correct (with regards to the full paper count), then the probability that the audit will mistakenly confirm the results is lower than some predetermined parameter, referred to as the *risk-limit* of the audit.

▶ **Definition 1.** ***The RLA Guarantee**: If the reported winners of the elections are not correct, an RLA will approve them w.p. of at most α, where α is a parameter which is set before the audit begins. α is referred to as the audit's **risk-limit**.*

The efficiency of an RLA is measured by the number of paper-ballots it requires to read, given that the reported tally matches the true one. In most cases, an RLA should remain relatively efficient even if the reported tally isn't completely accurate, as long as it results in the same winners as the true tally. The efficiency of any specific RLA method is limited by the election system it operates on. If a system has a sensitive social choice function, meaning that small tallying errors can often change the election winners, then it is more difficult to audit efficiently.

RLA methods generally belong to one of three categories, as defined by Lindeman and Stark [9]:

1. **Ballot-comparison:** In ballot-comparison audits, the auditor knows which paper-ballot matches which electronic-ballot. This category of audits is the most efficient, since it contains the most information about the election results.
2. **Ballot-polling:** In ballot-polling audits, a single paper-ballot can be sampled and examined, but it does not need to be matched to its corresponding electronic-ballot.
3. **Batch-level:** In batch-level audits, ballots are partitioned into batches, usually according to the prescient in which they were cast. The reported tally of each batch is available, but there's no guarantee that a paper-ballot in the batch can be matched to its electronic counterpart. Ballots are usually not randomly partitioned, and different batches are of different sizes. Batch-level audits are generally the least efficient of the three categories, since the partition into batches is not random, making it more difficult to get a representative sample of the overall vote distribution.

1.1 Our Contributions

The goal of the work is to expand the realm where RLAs are used. Its new contributions are:

1. A new and general method for performing **batch-level RLAs**, which can be applied for many election systems, is presented in Section 3. This method, which we call "*Batchcomp*", is usable for any election system that can be audited using the SHANGRLA framework [11].
2. An RLA method for the Israeli Knesset (The Israeli parliament) elections, based on the SHANGRLA framework, is presented in Section 4. This method can be applied as-is to conduct ballot-level RLAs, or be combined with Batchcomp to conduct a batch-level RLA.

To test both the Knesset RLA method and Batchcomp, we simulate their combination on real election results. While our Knesset RLA method is essentially a synthesis and adaptation of previous suggestions in the literature, it is the first time RLAs are applied to this setting.

3. A new type of RLA that applies to population censuses. This new type of audit is applicable in nations where political representatives are allocated to the nation's geographical regions based on their population, like the United States, Germany, Cyprus and more. It relies on data that is already collected in many countries, as part of an existing method for assessing the accuracy of population censuses called a "post enumeration survey" (PES). To the best of our knowledge, this is the first and only method which verifies the census' resulting allocation of representatives to federal-states with a clear statistical guarantee. The method is presented in Section 5.

2 Related Work

2.1 SHANGRLA

SHANGRLA [11] is an auditing framework which aids in adapting existing RLA algorithms to new social choice functions. It can be applied to a variety of election methods used globally, such as plurality, Hamiltonian elections [2], many proportional representation methods [1], and more.

This method is based on an abstraction called "sets of half-average nulls" (SHAN), where given a collection of finite lists containing unknown non-negative numbers, we wish to test whether the average of all of those lists is greater than $\frac{1}{2}$ by querying for the values at different indexes. Each query in this problem returns the values all lists hold at some specified index. An election system can be audited using SHANGRLA by reducing the problem of approving its reported winners to the SHAN problem. Once such a reduction is found, a number of existing algorithms [12, 11, 14] for the SHAN problem can be used to perform an RLA on that system.

This reduction is found by defining ℓ mappings $a_1, ..., a_\ell$ from the paper-ballots to non-negative values, such that the mean of every mapping across all backup paper-ballots is above $\frac{1}{2}$ iff the reported winner/s of the election are true.

▶ **Definition 2.** *The reported winners of an election system can be audited using SHANGRLA if there exist ℓ non-negative functions $a_1, ..., a_\ell$, called **assorters**, such that the reported winners of the elections are true iff for every $k \in [\ell]$:*

$$\frac{1}{|B|}\sum_{b\in B} a_k(b) > \frac{1}{2}, \tag{1}$$

*where B is a list of the backup paper-ballots of the elections. The ℓ inequalities above (for each $k \in [\ell]$) are referred to as the election's **assertions**.*

Some social choice functions have simple conversions to SHANGRLA assertions. E.g., a majority election between two candidates, Alice and Bob, can be audited using SHANGRLA with a single assorter. If Alice won the election according to the reported vote tally, this can be verified by using an assorter which has a mean greater than $\frac{1}{2}$ iff Alice truly received more votes than Bob:

▶ **Definition 3.** *An assorter which verifies that Alice received more votes from Bob in majority elections is:*

$$a(b) = \begin{cases} 1 & \textit{if } b \textit{ is for Alice} \\ 0 & \textit{if } b \textit{ is for Bob} \\ \frac{1}{2} & \textit{if } b \textit{ is invalid} \end{cases}$$

2.2 Finding SHANGRLA Assertions

In the example above, finding the correct assorter is relatively simple. For other election systems, which use more complicated social choice functions, verifying the correctness of the election winners can sometimes be reduced to verifying a set of linear inequalities regarding the various vote tallies. In such situations, it may not be immediately clear how to reduce them to assertions of the form $\frac{1}{|B|}\sum_{b\in B} a(b) > \frac{1}{2}$. For such cases, Blom et al. [1] suggests a generic solution. This solution reduces the problem of verifying that a set of linear inequalities that depend on the various vote tallies are all true to the problem of verifying that a set of assorters all have a mean greater than $\frac{1}{2}$ across all paper-ballots. We describe this solution for a single inequality. Given multiple inequalities, each inequality can be converted to a single SHANGRLA assertion in the same manner.
Say we have a linear inequality which is true iff the reported winner/s of some election system are the true ones:

$$\sum_{c\in\mathcal{C}} \beta_c v^{true}(c) > d, \tag{2}$$

where $\mathcal{C}$ is the set of all ballots that a single voter may cast (e.g. in plurality elections, $\mathcal{C}$ would be the set of candidates), $v^{true}(c)$ is the number of cast ballots of of type c according to the true results, and d and β_c (for each $c \in \mathcal{C}$) are constants. To perform an RLA for this election system, we wish to find a SHANGRLA assertion which is equivalent to (2). Meaning, given (2), we wish to find a non-negative function $a\colon C \to [0,\infty)$ such that (1) is equivalent to (2). As Blom et al. suggest, this can be achieved by defining:

$$a(b) := \frac{q - \beta_b}{2\left(q - \frac{d}{|B|}\right)}, \tag{3}$$

where $q := min_{c\in\mathcal{C}}\{\beta_c\}$, and β_b is determined by the type of ballot b is - if b is of type $c \in C$, we have $\beta_b = \beta_c$. As noted by Blom et al., the assorters generated by this method are non-negative as long as the inequality they are derived from isn't trivially true or trivially false, for any distribution of votes.

2.3 The ALPHA Martingale Test

The ALPHA Martingale Test [12] is a specific RLA algorithm for election systems which have a SHANGRLA reduction as described in Section 2.1. I.e., when there exist ℓ assorters $a_1, ..., a_\ell$ such that the reported winners of the elections are true iff for all $k \in [\ell]$ the inequality (1) is true.
The test operates by keeping ℓ variables $T_1, ..., T_\ell$, each representing the inverse of a p-value for the hypothesis that a certain list has an average greater than $\frac{1}{2}$. It then queries sequentially for random backup paper-ballots, where after each ballot it updates these ℓ variables. If at any point T_k for some $k \in [\ell]$ surpasses the threshold $\frac{1}{\alpha}$, it means

that we have sufficient evidence that the mean of its corresponding assorter a_k over all ballots is greater than $\frac{1}{2}$. If after a certain query, all of $T_1, ..., T_\ell$ have surpassed $\frac{1}{\alpha}$ at some point during the audit, then the reported winners of the elections are approved.

After each queried backup paper-ballot b_i, the algorithm updates T_k for every $k \in [\ell]$ by comparing $a_k(b_i)$ to the following values, which are set before b_i is revealed:

a. μ_k: The mean value of a_k over all ballots that have yet to be audited, given that the mean of a_k over all ballots is $\frac{1}{2}$. Recall that if the mean of a_k over all ballots is at most $\frac{1}{2}$, then the reported winners of the elections are wrong, which is the case the algorithm wishes to detect. Thus, if at some point during the audit we sample a ballot b with $a_k(b) \leq \mu_k$, it provides evidence that the reported winners of the elections are less likely to be correct, and vice-versa.

b. η_k: A guess for what we would expect $a_k(b_i)$ to be based on the reported results and the ballots we previously queried. This guess can be made in several ways while maintaining the algorithm's correctness. One reasonable way to do so is to set η_k to be the mean of a_k over ballots that have yet to be audited, assuming that the reported tally is completely accurate. As explained by Stark [12], The audit becomes more efficient, meaning less ballots need to be examined, the more accurate this guess is.

c. u_k: In the paper presenting ALPHA, u_k was defined as the maximal value a_k may return. In reality, the ALPHA Martingale Test is risk-limiting even for other choices of u_k, as long as the inequality $\mu_k < \eta_k < u_k$ is always maintained. For our purposes, u_k can be thought of as a guess for whether the next sampled ballot would indicate that assertion k is more or less likely to be true. If the next ballot to be sampled increases our confidence that the assertion is true, the audit is more efficient when u_k is large, and vice-versa.

The ALPHA Martingale Test can be adapted to sample ballots either with or without replacement. It can also be adapted to perform batch-level audits, where batches of ballots are sampled instead of individual ones. We refer to this batch-level version of the ALPHA Martingale Test as *ALPHA-Batch*. The Batchcomp method presented in Section 3 is based on ALPHA-Batch and attempts to improve on it by adjusting its assorters and utilizing the new definition for u_k.

3 The Batchcomp RLA

This section describes a generic way of performing batch-level RLAs, when the results of the elections can be verified using SHANGRLA assertions, as described in Section 2.1. This algorithm is original to this work and is based on ALPHA-Batch. Batchcomp relies on the following assumptions:

1. The election's social choice function can be audited using the SHANGRLA framework.
2. The reported and true results agree on the total number of ballots within each batch.

3.1 Model and Notation

Fix some elections system with a set of ballots B and a partition of these ballots into d batches $B_1, ..., B_d$. We make no assumptions regarding this partition, and different batches may be of different size. By assuming that the election system can be audited using SHANGRLA, we assume the following:

▶ **Assumption.** *There are ℓ assorters $a_1, .., a_\ell$ such that the reported winners are true iff for all $k \in [\ell]$:*

$$\frac{1}{|B|}\sum_{b\in B} a_k(b) > \frac{1}{2}.$$

Throughout the following sections, we sometimes abuse notation and apply assorters over entire batches. When doing so, $a_k(B_i)$ is defined as the mean of a_k over all ballots in B_i:

$$a_k(B_i) := \frac{1}{|B_i|}\sum_{b\in B_i} a_k(b). \tag{4}$$

In accordance with this, $a_k(B)$ denotes the mean value of a_k across all ballots.

Finally, note that each batch has a reported tally, which is known before the audit begins, and a true tally, which we may only learn during the audit. Therefore, each assorter has a reported and true mean value over each batch, which can be calculated from its reported and true tally, respectively. We denote the reported mean of an assorter a_k over a batch B_i as $a_k^{rep}(B_i)$, and its true mean over that batch as $a_k^{true}(B_i)$. Using this notation, the audit's goal is to test whether $a_k^{true}(B) > \frac{1}{2}$ for all $k \in [\ell]$.

3.2 Batchcomp Overview

Batchcomp attempts to confirm that the mean of ℓ assorters over all ballots are all greater than $\frac{1}{2}$ by sequentially sampling batches of backup paper-ballots and examining them. In each iteration, it samples a previously unsampled batch, such that each batch is sampled w.p. proportional to its size.

After each sampled batch, it updates ℓ p-values, each corresponding to the hypothesis that an assorter has a mean greater than $\frac{1}{2}$ across all ballots. The algorithm keeps the inverses of these p-values, $T_1, ..., T_\ell$. Each variable T_k is updated according to the backup paper-ballots in the sampled batch and according to 3 additional variables - μ_k, η_k, U_k. μ_k and η_k are defined as they were in the ALPHA Martingale Test (see Section 2.3). U_k, which is Batchcomp's version of u_k from the ALPHA Martingale Test, controls how significantly T_k changes per audited batch. μ_k, η_k and U_k are updated after each iteration, while always maintaining $U_k > \eta_k > \mu_k$.

During the audit, Batchcomp uses a modified version of the election's assorters $a_1, ..., a_\ell$. We denote these modified assorters as $A_1, ..., A_\ell$. Each new assorter A_k has a mean greater than $\frac{1}{2}$ iff its corresponding assorter a_k also has a mean which is greater than $\frac{1}{2}$. Thus, to approve that the reported winners of the elections are correct, it suffices to approve that $A_k(B) > \frac{1}{2}$ for all $k \in [\ell]$. Auditing $A_1, ..., A_\ell$ instead of $a_1, ..., a_\ell$ makes the audit agnostic to the order in which batches are sampled, as long as the reported batch-level vote tallies are accurate. As explained in the following section, this can increase the audit's efficiency.

3.3 Comparing Batchcomp and ALPHA-Batch

The ALPHA-Batch method, which Batchcomp is based on, is performed by examining the mean of every assorter over each sampled batch according to its backup paper-ballots. It does not use the reported vote tally of the batches beyond the total number of ballots they contain. Batchcomp attempts to improve on the efficiency of ALPHA-Batch by auditing something slightly different - instead of auditing the mean value of an assorter a_k over the backup paper-ballots (true results) in a sampled batch, it audits the discrepancy between the mean value taken by a_k over a batch according to its reported tally, and the mean value it returns over the same batch according to its paper-ballots.

The values returned by the ALPHA-Batch assorters can change drastically from batch to batch, depending on their vote distribution according to the true results. The values the Batchcomp assorters return depend only on the accuracy of the reported tally; if two batches with different vote distributions were both counted accurately in the reported results, a Batchcomp assorter will return the same value when applied on each of them. This fact is shown in Section 3.4.

As an example of this, examine majority elections with accurate reported tallies. In such elections, ALPHA-Batch operates by applying the assorter from Definition 3 on the sampled batches. Applying this assorter on a batch returns the share of votes won by the reported winner of the elections inside that batch. This value can swing heavily depending on the specific batch that is sampled. A batchcomp assorter for the same elections returns the same value on every batch, regardless of the vote distribution within it.

Recall that before sampling and reading a backup paper-ballot, the ALPHA Martingale Test guesses the value that each assorter would return on this ballot (this guess is η_k, for each assorter a_k). As explained by Stark when presenting ALPHA [12], the audit is more efficient when these guesses are accurate. If each assorter returns a similar value for all batches, as happens in Batchcomp, then the audit can make guesses which are more accurate. This is the root cause for Batchcomp outperforming ALPHA-Batch in the simulations shown in Section 4.3.

3.4 The Batchcomp Assorters

This section converts the election assorters $a_1, ..., a_\ell$ to equivalent assorters $A_1, ..., A_\ell$ which depend on the accuracy of the batch-level tallies instead of their vote distribution. These new assorters, which we refer to as the *Batchcomp assorters*, are equivalent to the original ones in the sense that they all have a mean greater than $\frac{1}{2}$ iff the original ones all have a mean greater than $\frac{1}{2}$.

▶ **Definition 4.** *For each assorter a_k, define the **Batchcomp-assorter** $A_k : C^* \to [0, \infty)$:*

$$A_k(B_i) := \frac{1}{2} + \frac{M_k + a_k^{true}(B_i) - a_k^{rep}(B_i)}{2(w_k - M_k)}.$$

Where M_k is the reported margin of assorter a_k across all batches, and w_k is the maximal reported value of a_k, across all batches:

$$M_k := a_k^{rep}(B) - \frac{1}{2}, \quad w_k := \max_{j\in[d]}\{a_k^{rep}(B_j)\}.$$

As explained in Section 3.3, when the reported batch-level tallies are accurate, each Batchcomp assorter returns the same value on all batches. This is since accurate batch-level tallies indicate that for any batch B_i we have $a^{rep}(B_i) = a^{true}(B_i)$, and:

$$A_k(B_i) = \frac{1}{2} + \frac{M_k}{2(w_k - M_k)}.$$

To use these Batchcomp assorters instead of the original assorters $a_1, ..., a_\ell$, we need to show that they are non-negative and that $A_k(B) > \frac{1}{2}$ iff $a_k^{true}(B) > \frac{1}{2}$ (recall that $a(B)$ denotes the mean of an assorter a over all ballots).

▷ Claim 5. For any assorter a_k, its conversion to a Batchcomp assorter A_k is non-negative.

Proof. Fix an assorter a_k and its Batchcomp counterpart A_k. Examine the minimum of a_k^{true} and the maximum of a_k^{rep}. Recall that assorters are always non-negative, and that w_k is defined as the maximum of a_k^{rep} across all batches. Thus, for any batch B_i:

$$A_k(B_i) = \frac{1}{2} + \frac{M_k + \overbrace{a_k^{true}(B_i)}^{\geq 0} - \overbrace{a_k^{rep}(B_i)}^{\leq w_k}}{2(w_k - M_k)} \geq \frac{1}{2} + \frac{M_k - w_k}{2(w_k - M_k)} = 0. \qquad \triangleleft$$

▷ Claim 6. For any assorter a_k and its conversion to a Batchcomp assorter A_k, we have $a_k^{true}(B) > \frac{1}{2}$ iff $A_k(B) > \frac{1}{2}$.

Proof. By the definition of A_k and M_k (Definition 4):

$$\begin{aligned} A_k(B) &= \frac{1}{2} + \frac{M_k + a_k^{true}(B) - a_k^{rep}(B)}{2(w_k - M_k)} \\ &= \frac{1}{2} + \frac{a_k^{rep}(B) - \frac{1}{2} + a_k^{true}(B) - a_k^{rep}(B)}{2(w_k - M_k)} \\ &= \frac{1}{2} + \frac{a_k^{true}(B) - \frac{1}{2}}{2(w_k - M_k)}. \end{aligned}$$

And since $w_k > M_k$, as $w_k \geq a_k^{rep}(B) > M_k$, this value is greater than $\frac{1}{2}$ iff $a_k^{true}(B) > \frac{1}{2}$. ◁

The Batchcomp assorters $A_1, ..., A_\ell$ can also be used by the ALPHA-Batch algorithm in place of the original assorters $a_1, ..., a_\ell$. This, however, does not lead to an increase in the audit's efficiency by itself, at least in the settings we simulated. Batchcomp attempts to improve on ALPHA-Batch's efficiency by combining these new assorters with the re-definition of u_k (see Section 2.3).

3.5 The Batchcomp Algorithm

1. **Initialization:**
   (a) Initialize $\mathcal{K} = [\ell]$, which holds the indexes of assertions we have yet to approve.
   (b) Initialize $\mathcal{B}^1 = (B_1, B_2, ..., B_d)$ and $\mathcal{B}^0 = \emptyset$. As the algorithm progresses, $\mathcal{B}^0$ holds the batches which were already audited and $\mathcal{B}^1$ the batches that have yet to be audited.
   (c) For each $k \in \mathcal{K}$ initialize:

   $$T_K := 1, \quad \mu_j := \frac{1}{2}, \quad \eta_k := \frac{1}{2} + \frac{M_k}{2(w_k - M_k)}, \quad U_k := \frac{1}{2} + \frac{M_k + \delta}{2(w_k - M_k)}.$$

   For some $\delta > 0$. Appendix B examines how to choose δ. For definitions of M_k and w_k see Definition 4. Note that since $w_k > M_k > 0$ we have $U_k > \eta_k > \mu_k$.
2. **Auditing Stage:** As long as $\mathcal{B}^1 \neq \emptyset$, perform:
   (a) Sample a batch from $\mathcal{B}^1$ and denote it as B_i. Each batch B_j in $\mathcal{B}^1$ is sampled with probability proportional to its size: $\frac{|B_j|}{\sum_{B_t \in \mathcal{B}^1} |B_t|}$.
   (b) Remove B_i from $\mathcal{B}^1$ and add it to $\mathcal{B}^0$.
   (c) For each $k \in K$, update T_k by the same update rule as in ALPHA-Batch:

   $$T_k \leftarrow T_k \left(\frac{A_k(B_i)}{\mu_k} \frac{\eta_k - \mu_k}{U_k - \mu_k} + \frac{U_k - \eta_k}{U_k - \mu_k} \right).$$

   (d) For each $k \in \mathcal{K}$, if $T_k > \frac{1}{\alpha}$, the kth assertion can be approved, so remove k from $\mathcal{K}$.

(e) For each $k \in \mathcal{K}$ update u_k, μ_k and η_k, in this order:
- $\mu_k \leftarrow \frac{\frac{1}{2}n - \sum_{B_j \in \mathcal{B}^0} |B_j| A_k(B_j)}{n - \sum_{B_j \in \mathcal{B}^0} |B_j|}$.
- $\eta_k \leftarrow max\left\{\frac{1}{2} + \frac{M_k}{2(w_k - M_k)}, \mu_k + \epsilon\right\}$.
- $U_k \leftarrow max\left\{U_k, \eta_k + \epsilon\right\}$.

Where ϵ is some very small positive meant to ensure that $\mu_k < \eta_k < U_k$.

(f) If $\mu_k < 0$, The kth assertion is necessarily true, so remove k from $\mathcal{K}$.

(g) If $\mathcal{K} = \emptyset$, all assertions were approved, so approve the reported winners.

3. **Output:** If the audit hasn't approved after examining all batches, it can calculate the true winners of the elections.

Any initialization and update rule for the variables η_k and U_k that always maintains $\mu_k < \eta_k < U_k$ also yields a risk-limiting audit. The update rules shown here lead to increased efficiency when the batch-level tallies are accurate. η_k, the algorithm's guess for the value A_k would return on the next sampled batch, is set to the value A_k returns on each batch given that the reported batch-level tallies is accurate, as calculated in Section 3.4.

▶ **Theorem 7.** *Batchcomp fulfills the RLA guarantee (Definition 1).*

Proof. Batchcomp is a modified version of ALPHA-Batch, and fulfills the RLA guarantee for the same reasons as ALPHA-Batch. It makes two modifications to the ALPHA-Batch algorithm, which maintain it being risk-limiting:

1. For every $k \in [\ell]$, Batchcomp verifies that $A_k(B) > \frac{1}{2}$ while ALPHA-Batch verifies that $a_k^{true}(B) > \frac{1}{2}$. By Claim 6, verifying these two conditions is equivalent. ALPHA-Batch also relies on $a_1, ..., a_\ell$ being non-negative. Switching to auditing $A_1, ..., A_\ell$ requires them to be non-negative as well, which is proven in Claim 5.
2. Batchcomp uses a different initialization and update rule for U_k. While ALPHA-Batch defines U_k differently than Batchcomp, it only requires to have $U_k > \eta_k$ for every $k \in [\ell]$ for the audit to fulfill the RLA guarantee. Batchcomp's update rule for U_k and η_k (step 2e) always maintains $U_k > \eta_k$, meaning that it fulfills the guarantee as well. ◀

4 Israeli Knesset Elections RLA

This section describes how to perform an RLA to verify the results of the Israeli Knesset elections. The Knesset is the Israeli parliament and its sole legislative authority. It comprises of 120 members who are elected according to closed party-list proportional representation. The goal of this suggested Knesset RLA is to verify that each party won the correct number of seats, meaning that the correct Knesset members were elected.

This method can be used in Israel currently to verify the initial hand-count of the votes, which is not performed centrally - each polling place independently tallies its own ballots. It can also become useful if, in the future, the vote tallying will be done by some electronic means, such as an optical reader. In such cases, this method could confirm that the winners outputted by the electronic vote tabulation system are likely to be correct.

Before moving to explain the social choice function of the Knesset elections, we define some notation. Let P be the set of all parties running in the elections, and let $S := 120$ be the number of available seats. For every party $p \in P$, let $v^{true}(p)$ denote the true number of votes p received, according to the backup paper-ballots.

4.1 Knesset Election Method

Before each election cycle, each running party submits a ranked list of its candidates. On polling day, each voter votes for a single party, and parties receive seats in proportion to the share of the votes they received. The seats each party wins are given to the top-ranked candidates in the party's list. Allocating Knesset seats to the various parties is done as follows [8]:

Electoral Threshold: In the Knesset elections, only parties who receive a share of at least $t := 0.0325$ of the valid votes are eligible to win seats.

Seat allocation: The allocation of seats is done according to the D'Hondt method, a highest averages method, and can be formulated in multiple ways. We present a description of a general highest averages method which was suggested previously by Gallagher [5]. Each specific highest averages method is characterized by a unique monotonically increasing function $d : \mathbb{N} \to \mathbb{N}$ which is used during the seat allocation process. D'Hondt, the method used in the Israeli Knesset, uses $d(n) = n$. To find how many seats a party is awarded for a highest averages method with some function d:

1. Imagine a table with a row for each party which is above the threshold, and S columns. At column s In the row of party p, write $v^{true}(p)/d(s)$. All cells are initially unmarked.
2. Mark the S cells with the largest values in the table.
3. The number of marked cells a party has in its row is the number of seats it receives.

Note that the values in each row are monotonically decreasing, as d is monotonically increasing, so each row would be fully marked up to a certain column, and unmarked for the rest of it.

Apparentment (Also Known as Electoral Alliances): Prior to election day, two parties may sign an apparentment agreement, which may allow one of them to gain an extra seat. If two parties sign an apparentment agreement, and only if both are above the threshold, they unite to a single allied party during the seat allocation stage. Then, the number of seats their alliance received is split between them according to the same seat allocation method (D'Hondt). If one of the parties in the apparentment is below the electoral threshold while using only its own votes, the apparentment is ignored. Each party may only sign a single apparentment agreement.

4.2 Knesset RLA Assorters

This section presents assorters that can be used to perform an RLA for the Knesset elections, using the SHANGRLA framework. We begin by presenting three conditions which all hold true iff the reported winners of the elections are correct. We then proceed to show assorters for these conditions, such that the assorters all have a mean greater than $\frac{1}{2}$ iff these conditions all hold true.

▶ **Theorem 8.** *Let $s^{rep}(p)$ and $s^{true}(p)$ be the reported and true number of seats that a party p won in a Knesset elections, respectively. We have it that $s^{rep}(p) = s^{true}(p)$ for every party $p \in P$, iff these 3 conditions all hold true:*

1. *Every party who is reportedly above the electoral threshold, is truly above it.*
2. *Every party who is reportedly below the electoral threshold, is truly below it.*
3. *For every two parties p_1, p_2 who are reportedly above the electoral threshold, the condition $(s^{rep}(p_1) \geq s^{true}(p_1)) \vee (s^{rep}(p_2) \leq s^{true}(p_2))$ is true.*

Proof. Fix some reported and true tallies for the elections, and calculate the number of seats each party reportedly and truly won according to these tallies. If the reported and true number of seats each party won are equal, then the 3 conditions above hold true trivially.

Otherwise, assume there is a discrepancy between the reported and true seat allocation. There must be at least one party who won more seats according to the reported results compared to the true results, which we denote as p_r, and at least one party who won less seats according to the reported results compared to the true results, which we denote as p_t.

We now show that at least one of the three conditions above are violated. If p_r is not truly above the electoral threshold, then Condition 1 is violated, as it receives seats according to the reported tally. Similarly, if p_t is below the threshold according to the reported tally, then Condition 2 is violated. Otherwise, both parties are truly above the threshold.

If both parties are reportedly above the electoral threshold, then p_t reportedly won less seats than it truly deserves, meaning that $s^{rep}(p_t) < s^{true}(p_t)$. Similarly, we have $s^{rep}(p_r) > s^{true}(p_r)$. This violates Condition 3 and concludes the proof. ◀

Above Threshold Assertion

The role of this assertion is to check that Condition 1 holds. Stark [11] has previously suggested a SHANGRLA assertion for this condition exactly. For every party p who reportedly is above the electoral threshold, we add a single SHANGRLA assorter to the set we audit:

▶ **Definition 9.** *An above threshold assorter for a party p is defined as:*

$$a_p^{above}(b) := \begin{cases} \frac{1}{2t} & \textit{if } b \textit{ is for party } p \\ \frac{1}{2} & \textit{if } b \textit{ is invalid} \\ 0 & \textit{otherwise} \end{cases}$$

Below Threshold Assertion

This assertion verifies Condition 2. Confirming that a party is below the threshold is equivalent to verifying that all other parties received at least $1 - t$ of the valid votes. Therefore, we can use an assorter similar to the one above. For every party p who is reportedly below the electoral threshold, we add the following assorter to the set we audit:

▶ **Definition 10.** *A below-threshold assorter for party p is defined as:*

$$a_p^{below}(b) := \begin{cases} 0 & \textit{if } b \textit{ is for party } p \\ \frac{1}{2} & \textit{if } b \textit{ is invalid} \\ \frac{1}{2(1-t)} & \textit{otherwise} \end{cases}$$

Move-Seat Assertion

This assertion is verifies that Condition 3. For any pair of parties p_1, p_2, this essentially verifies that compared to the reported results, p_1 does not deserve extra seats at the expense of p_2. An assertion for this condition was previously suggested by Blom et al. [1] (Section 5.2.) when auditing elections using highest averages methods. For every ordered pair of parties (p_1, p_2) who are reportedly above the electoral threshold, we add the following assorter to the set we audit:

▶ **Definition 11.** *A move-seat assorter for two parties p_1, p_2 is defined as:*

$$a_{p_1,p_2}^{move}(b) := \begin{cases} \frac{1}{2} + \frac{s^{rep}(p_1)+1}{2s^{rep}(p_2)} & \textit{if } b \textit{ is for } p_2 \\ 0 & \textit{if } b \textit{ is for } p_1 \\ \frac{1}{2} & \textit{otherwise} \end{cases}$$

Handling Apparentments

The assertions above ignore the existence of apparentments. To handle them, we can simply treat each two allied parties who are reportedly above the electoral threshold as a united party when adding move-seat assertions. Additionally, to verify that the seat allocation between every two allied parties is correct, two move-seat assertions (one in each direction) are added to the audit for every two allied parties who are reportedly above the electoral threshold.

4.3 Simulations Based on Recent Elections

We describe the results of simulating the execution of a batch-level RLA over the real election results for the 24th Knesset in 2021. The partition of ballots to batches used in this simulation is done according to the real election results, and each batch contains ballots from a single polling place. The audit uses assertions as described in Section 4.2, converts their assorters to Batchcomp assorters as described in Section 3.4 and runs the Batchcomp method described in Section 3.5 to audit them.

We compare the performance of Batchcomp with the performance of the ALPHA-Batch algorithm described in section 4.2 of ALPHA [12]. ALPHA-batch uses the SHANGRLA assertions from Section 4.2 of this work, without converting their assorters to Batchcomp-assorters. For each assertion, we measure the number of ballots required to approve it by each algorithm, as a factor of the assertion's margin (minimal number of ballots that would need to be altered, compared to the reported vote tally for the assertion to become false).

The results presented here assume that *all vote tallies are accurate.* Similar plots for results with small tabulation errors, as well as results for additional election cycles, are available in the paper's GitHub repository. The election cycle described here is representative of the trends present in the other examined cycles.

Technical Details

The simulated RLA uses a risk-limit of $\alpha = 0.05$ and $\delta = 10^{-10}$. The latter was determined after some experimentation - lower choices for δ do not improve efficiency when the reported results are accurate, while higher values reduce the audit's efficiency.

The number of audited ballots by each method is averaged across 10 simulations. An examination of these simulations shows that the number of ballots required to approve each assertion by Batchcomp has very low standard deviation. The mean standard deviation across all assertions is 1,888, while the maximal standard deviation across all assertions is 5,291. The code used to generate these simulations was written in Python, and is available at the paper's GitHub repository (see title page), along with plots for additional election cycles.

Results

Figure 1 and Table 1 show that approving the reported winners for this election cycle required auditing 85% of ballots by Batchcomp, while requiring virtually all ballots by ALPHA-Batch. If it wasn't for a single assertion which had a very small margin (367 ballots), the Batchcomp audit would be done after auditing $\tilde{3}2\%$ of the ballots, while ALPHA-Batch would still require reading nearly all ballots.

The most glaring conclusion from this simulation, as well as ones we performed for additional election cycles, is that Knesset elections have very tight margins, which make them difficult to audit in a risk-limiting manner. If the election winners win with a margin

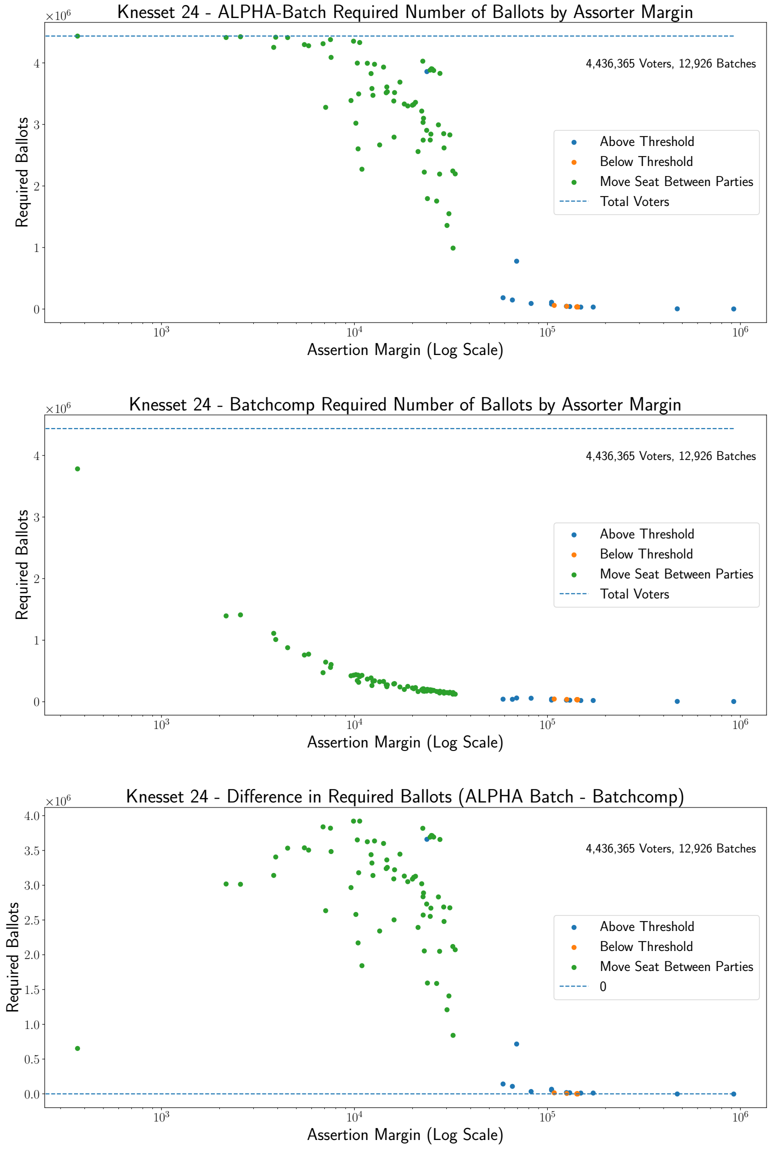

Figure 1 The first two plots present the number of ballots required to approve each assertion during the audit, either by the ALPHA-Batch method or by our Batchcomp method. Each point in these plots represents a single assertion, where its value on the x axis is its margin in log-scale, and its value on the y axis is the number of ballots that the audit examined before approving the assertion. Each point in the plot is colored by the type of assertion it represents. The final plot presents the difference in ballots required per assertion between ALPHA-Batch and Batchcomp.

of below 0.01% of the total ballots, it's unlikely that any RLA method could approve them without close to a full manual recount. Appendix D examines ways of relaxing the RLA's guarantee to decrease the number of ballots the audit has to read.

While auditing the entire Knesset elections proves to be difficult, examining the number of ballots required to approve the various assertions shows that Batchcomp significantly outperforms ALPHA-Batch. Generally, assertions that had very small or fairly large margins required a similar number of ballots by both algorithms, while assertions with margins of

Table 1 The last three assertions to be approved by the Batchcomp, including their margin and the number of ballots they required to be approved by each method.

Assertion	Margin (% of votes)	Batchcomp (% of votes)	ALPHA (% of votes)
Don't move a seat from Meretz to Labor	367 (0.008%)	3,782,269 (85%)	4,435,198 (≈ 100%)
Don't move a seat from The Joint List to Likud & Religious Zionist	2,567 (0.06%)	1,411,262 (32%)	4,424,877 (≈ 100%)
Don't move a seat from New Hope to Yamina	2,162 (0.05%)	1,394,595 (31%)	4,412,625 (99%)

between 0.01% and 2% were significantly easier to audit using Batchcomp. Some assertions which ALPHA-Batch could not approve without a nearly full manual recount were approved by Batchcomp while examining less than 20% of the backup paper-ballots.

5 The Census RLA

This section presents a risk-limiting audit method for a population census. It applies to nations which allocate political power to their constituencies or federal-states in proportion to their population according to a certain class of methods (highest averages), and who conduct a post-enumeration survey (PES) as recommended by United Nations guidelines [13]. By these guidelines, a PES is performed by randomly sampling a small number of households, re-running the census over this chosen sample, and then comparing the results to the original census. For consistency, throughout this section, we assume that this allocated political power is manifested as the number of representatives a region receives in parliament, and refer to these regions as the nation's federal-states.

The goal of our audit is to provide a clear statistical guarantee regarding the correctness of this census' resulting allocation of representatives. To achieve such a guarantee, we first need to define what allocation is considered correct. The allocation which results from the PES would not be sufficient here, since it may change based on the subset of households which were sampled. To avoid this potential issue, we view the true results of the census as the results the PES would have if it was to run over all households. This means that technically, a census RLA assumes that the PES surveyed all households. During the actual audit, however, it only asks for the information the PES collected on a small, randomly chosen sample of households, which is exactly the data that the PES actually has.

The census RLA is performed by sequentially sampling households and processing the census and PES information regarding them. Since the PES only runs over a small sample of households, the audit is limited in its length. Therefore, setting a risk-limit (probability of approving wrong results) ahead of the audit, as done in election RLAs, could be problematic. Were we to do so, then the audit might fail to approve a correct representative allocation even when using the entire PES sample, resulting in an inconclusive outcome.

The observation above leads us to slightly change the statistical guarantee that a census RLA provides: instead of setting the risk-limit and then running the audit, the census RLA runs over the entire PES and then returns the risk-limit with which it can approve the census representative allocation. If the risk-limit returned by census RLA is insufficient, a governing body may decide to conduct a second round of re-surveying, and to continue the audit on these newly re-surveyed households.

▶ **Definition 12.** ***The census RLA guarantee:*** *For any $0 < \alpha \leq 1$, if running the PES over all households would lead to a different allocation of representatives than the census, then the probability that a census RLA returns a value α' such that $\alpha' \leq \alpha$ is at most α. α' is referred to as the audit's* outputted risk-limit*.*

5.1 Post Enumeration Survey (PES)

A post enumeration survey is a process which measures the accuracy of a population census by conducting an independent population survey over a small portion of randomly chosen households. According to the guidelines published by the Department of Economic and Social Affairs of the United Nations [13], a PES begins by choosing a partial sample of the households in a nation, such that each household has an equal probability of being included in this sample. Afterwards, a new survey contacts each sampled household and asks them the exact same questions as the original census.

For our purposes, the only information of interest is the number of residents at each household. In reality, some countries may allocate representatives to federal-states according to the number of a specific sector of the population that they hold (e.g. eligible voters or citizens). In our model, we assume it is simply the number of residents, but our method applies in the same manner otherwise.

5.2 Model and Notation

In our model, a nation measures its population using a nation-wide census and then conducts a PES as described in the previous section. Denote the information given by the census as:

- H: A list of households that were surveyed.
- $g^{cen}(h)$: The number of residents a household $h \in H$ has according to the census.

And denote the information given by the PES as:

- H^{PES}: A list of households which were surveyed by the PES. Must be a subset of H.
- $g^{PES}(h)$: The number of residents a household $h \in H^{PES}$ holds according to the PES.

The nation then allocates R representatives to its federal-states, whose set we denote as $\mathcal{S}$, by using a *highest averages method*, as described in Section 4.1. Recall that each specific highest averages method is defined by a different monotonically increasing function $d : \mathbb{N} \to \mathbb{N}$.

Our model assumes each state also has a constant additive factor which is added to its census population count during the representative allocation process. We denote this constant as c_s for each $s \in \mathcal{S}$. Meaning, the value written at cell $[s, r]$ of the imaginary table used during the representative allocation, for $s \in \mathcal{S}$ and $r \in [R]$, is:

$$\frac{g^{cen}(s) + c_s}{d(r)}, \tag{5}$$

The additive factor, c_s, allows our model some added flexibility, meaning it can cover more political systems. In the United States, for example, we would want to exclude people living in group residence (e.g. homeless people, nursing home residents, etc') from the audit, since they are not covered by the PES. To do so, we can assume their number according to the census is accurate and run the audit over the rest of the population. This can be achieved by defining c_s to be the number of persons who live in a group residence in state s according to the census.

Our census RLA method relies on one simplifying assumption:

▶ **Assumption.** *In both the census and in the PES, the number of residents in a single household is upper-bounded by a known value, denoted as g^{max}.*

The value g^{max} must be set before the PES is conducted. Both the census and the PES must report that all households have g^{max} residents at most.

This assumption is necessary due to a critical difference between elections and censuses; In elections, a single ballot has very limited power. In a census, if it was not for this assumption, a single household could hold an arbitrarily large number of residents and completely swing the allocation of representatives to the states.

Finally, denote the number of representatives awarded to state $s \in \mathcal{S}$ according to the census as $r^{cen}(s)$.

5.3 Census RLA Overview

The following sections suggest a new method for census RLAs, which relies on the SHANGRLA framework. In the following section, we design SHANGRLA assertions for auditing the census' resulting allocation of representatives to the federal-states. While these assertions can be used as-is to perform a census RLA, they are only an intermediate step in the development of more efficient assertions. These more efficient assertions are used by a modified version of the ALPHA Martingale Test to perform a census RLA, as described in Section 5.5.

5.4 Census RLA Assorters

We begin by adapting the definition of assertions and assorters to the language of census RLAs. When auditing elections, an assorter is defined as a non-negative function over the set of possible ballots a voter may cast. When auditing a census, we define an assorter as a non-negative function over the set of all households, meaning $a\colon H \to [0,\infty)$. An assorter a satisfies the assertion $\frac{1}{|H|}\sum_{h\in H} a(h) > \frac{1}{2}$ iff some condition regarding the allocation of representatives to the federal states is true.

▶ **Definition 13.** ***Census assorters*** *are functions $a_1, ..., a_\ell : H \to [0,\infty)$ with the following property: Given some census results, if the PES surveyed all households, the allocation of representatives according to the census and according to the PES match iff for all $k \in [\ell]$:*

$$\frac{1}{|H|}\sum_{h\in H} a_k(h) > \frac{1}{2}. \tag{6}$$

These ℓ inequalities are referred to as the ***census assertions****.*

The census assorters for our setting are developed by finding a set linear inequalities which all hold true iff a full PES leads to the same allocation of representatives as the census. These inequalities are then converted to SHANGRLA style assertions by the method described by Blom et al. [1] (see Section 2.1).

▶ **Theorem 14.** *Assume the PES surveyed all households. The allocation of representatives according to the census and according to the PES match, iff for any two states $s_1, s_2 \in \mathcal{S}$:*

$$\frac{\sum_{h\in H} g^{PES}_{s_1}(h) + c_{s_1}}{d(r^{cen}(s_1))} > \frac{\sum_{h\in H} g^{PES}_{s_2}(h) + c_{s_2}}{d(r^{cen}(s_2)+1)}. \tag{7}$$

The proof of this theorem appears in Appendix A.1. By the method suggested by Blom et al. [1], confirming Equation (7) is equivalent to confirming the SHANGRLA style assertion:

$$\frac{1}{|H|}\sum_{h\in H} a^{PES}_{s_1,s_2}(h) > \frac{1}{2},$$

where:

▶ **Definition 15.** *The census assorter $a^{PES}_{s_1,s_2}$ is defined as:*

$$a^{PES}_{s_1,s_2}(h) := \frac{g^{PES}_{s_1}(h)}{cd(r^{cen}(s_1))} + \frac{g^{max} - g^{PES}_{s_2}(h)}{cd(r^{cen}(s_2)+1)},$$

where c denotes:

$$c := 2\left(\frac{g^{max}}{d(r^{cen}(s_2)+1)} - \frac{c_{s_1}}{|H|d(r^{cen}(s_1))} + \frac{c_{s_2}}{|H|d(r^{cen}(s_2)+1)}\right).$$

▶ **Theorem 16.** *Assume that the PES surveyed all households. The assorters $\{a^{PES}_{s_1,s_2} \mid s_1, s_2 \in \mathcal{S}, s_1 \neq s_2\}$ are all non-negative and satisfy the following condition: The allocation of representatives according to the census and the PES match iff for all $s_1, s_2 \in \mathcal{S}$:*

$$\frac{1}{|H|}\sum_{h\in H} a^{PES}_{s_1,s_2}(h) > \frac{1}{2}. \tag{8}$$

Proof. The non-negativity of these assorters is due to the method by Blom et al., which generates non-negative assorters. Additionally, by this method, for any two states s_1, s_2, verifying (8) is equivalent to verifying (7). By Theorem 14, verifying (7) for every two states is equivalent to verifying that the full PES leads to the same representative allocation as the census, concluding this proof. ◀

For each assorter $a^{PES}_{s_1,s_2}$, we now define a new assorter A_{s_1,s_2} which can also be used to audit the same census. A_{s_1,s_2} has a significant advantage over $a^{PES}_{s_1,s_2}$, which motivates us to use it instead. Each assorter $a^{PES}_{s_1,s_2}$ essentially audits the number of residents per household according to the PES, without using the per-household census data. Meanwhile, A_{s_1,s_2} audits the discrepancy in the number of household members between the census and the PES. Since we typically expect this discrepancy to be small, this yields a more stable and efficient audit.

Before defining A_{s_1,s_2}, note that each assorter $a^{PES}_{s_1,s_2}$ can also be defined over the census population counts instead of the PES counts. We denote this as $a^{cen}_{s_1,s_2}$:

▶ **Definition 17.** *The value of an assorter $a^{PES}_{s_1,s_2}$ as in Definition 15 over the census population count is defined as:*

$$a^{cen}_{s_1,s_2}(h) := \frac{g^{cen}_{s_1}(h)}{cd(r^{cen}(s_1))} + \frac{g^{max} - g^{cen}_{s_2}(h)}{cd(r^{cen}(s_2)+1)}.$$

Using this reported (by the census) and true (by the PES) resident counts, we define new assorters which audit the discrepancy between them. This is similar to the Batchcomp assorters from Section 3.4, which audit the batch-level discrepancy between the reported and true vote tallies.

▶ **Definition 18.** *The census comparison assorter A_{s_1,s_2} for states $s_1, s_2 \in \mathcal{S}$ is defined as:*

$$A_{s_1,s_2}(h) := \frac{1}{2} + \frac{m_{s_1,s_2} + a^{PES}_{s_1,s_2}(h) - a^{cen}_{s_1,s_2}(h)}{2(z_{s_1,s_2} - m_{s_1,s_2})},$$

where m_{s_1,s_2} is the margin of a_{s_1,s_2} according to the census population counts:

$$m_{s_1,s_2} := \frac{1}{|H|}\sum_{h\in H} a^{cen}_{s_1,s_2}(h) - \frac{1}{2},$$

and:

$$z_{s_1,s_2} := \max\left\{\frac{g^{max}}{cd(r^{cen}(s_2)+1)}, \frac{g^{max}}{cd(r^{cen}(s_1))}, 0\right\}.$$

▶ **Theorem 19.** *Assume that the PES surveyed all households. The assorters $\{A_{s_1,s_2} \mid s_1, s_2 \in \mathcal{S}\}$, as defined in Definition 18, are all non-negative and satisfy the following condition: the allocation of representatives according to the census and the PES match iff for all $s_1, s_2 \in \mathcal{S}$:*

$$\frac{1}{|H|}\sum_{h\in H} A_{s_1,s_2}(h) > \frac{1}{2}.$$

The proof for this theorem is presented in Appendix A.2.

5.5 Census RLA Algorithm

The algorithm presented next is a slightly altered version of the ALPHA Martingale Test, when thinking of each household as a ballot whose content is the household's state and its number of residents. We denote the households surveyed by the PES as $H^{PES} = (h_1, h_2, ..., h_d)$ for some $d \in \mathbb{N}$, and assume that they are given in random order.

1. **Initialization**
   (a) For each $(s_1, s_2) \in \mathcal{S} \times \mathcal{S}$ s.t. $s_1 \neq s_2$, initialize:
      - $T_{s_1,s_2} := 1$.
      - $T^{max}_{s_1,s_2} := 1$.
      - $\mu_{s_1,s_2} := \frac{1}{2}$.
      - $\eta_{s_1,s_2} := \frac{1}{2} + \frac{m_{s_1,s_2}}{2(z_{s_1,s_2} - m_{s_1,s_2})}$.
      - $U_{s_1,s_2} := \frac{1}{2} + \frac{m_{s_1,s_2}+\delta}{2(z_{s_1,s_2} - m_{s_1,s_2})}$, where $\delta > 0$.
2. **Auditing Stage:** Iterate over the households in H^{PES}, for the ith household h_i perform for each ordered pair of states (s_1, s_2):
   (a) Update T_{s_1,s_2} and $T^{max}_{s_1,s_2}$:
      - $T_{s_1,s_2} \leftarrow T_{s_1,s_2}\left(\frac{A_{s_1,s_2}(h_i)}{\mu_{s_1,s_2}}\frac{\eta_{s_1,s_2}-\mu_{s_1,s_2}}{U_{s_1,s_2}-\mu_{s_1,s_2}} + \frac{U_{s_1,s_2}-\eta_{s_1,s_2}}{U_{s_1,s_2}-\mu_{s_1,s_2}}\right)$.
      - $T^{max}_{s_1,s_2} \leftarrow \max\left\{T^{max}_{s_1,s_2}, T_{s_1,s_2}\right\}$.

   (b) Update $\mu_{s_1,s_2}, \eta_{s_1,s_2}$ and U_{s_1,s_2}, in this order:
      - $\mu_{s_1,s_2} \leftarrow \frac{\frac{1}{2}|H| - \sum_{j=1}^{i} A_{s_1,s_2}(h_j)}{|H|-i}$.
      - $\eta_{s_1,s_2} \leftarrow max\left\{\frac{1}{2} + \frac{m_{s_1,s_2}}{2(z_{s_1,s_2}-m_{s_1,s_2})}, \mu_{s_1,s_2} + \epsilon\right\}$.
      - $U_{s_1,s_2} \leftarrow max\{U_{s_1,s_2}, \eta_{s_1,s_2} + \epsilon\}$.

      Where ϵ is some very small positive meant to ensure that $\mu_{s_1,s_2} < \eta_{s_1,s_2} < U_{s_1,s_2}$.

   (c) For each s_1, s_2, if $\mu_{s_1,s_2} < 0$, the corresponding assertion must be true, so set $T^{max}_{s_1,s_2} = \infty$.
3. **Output:** The result of the audit is the maximal risk-limit across all assertions:

$$\max_{s_1,s_2\in\mathcal{S}}\left\{\frac{1}{T^{max}_{s_1,s_2}}\right\}.$$

▶ **Theorem 20.** *The census RLA fulfills the census RLA guarantee (Definition 12).*

Proof. The census RLA is essentially the ALPHA Martingale Test, with four modifications. We explain why these modifications maintain the risk-limiting nature of the ALPHA Martingale Test:

- Instead of sampling and examining ballots, the census RLA samples and examines households. This does not effect the fact that the ALPHA Martingale Test is risk-limiting - a census RLA can be viewed as a classical election RLA where every ballot correspond to a household, and holds that household's state and number of residents.

- The census RLA doesn't sample households at random, it iterates over the households sampled by the PES. Despite this, since the PES surveys randomly selected households, the algorithm audits a previously unsampled household selected uniformly at random in each iteration. This is just as the ALPHA Martingale Test requires.
- Instead of pre-setting the risk-limit, the risk-limit with which the census representative allocation can be approved is outputted after iterating over all PES households. This outputted risk-limit is already available as part of the ALPHA Martingale Test. In election RLAs, the audit approves the reported winners when this running risk-limit drops below the pre-set risk-limit. Here it outputs it after examining all PES households.
- The census RLA defines U_{s_1,s_2} (which corresponds to u_k in the ALPHA Martingale Test) differently. As mentioned previously, it always maintains $U_{s_1,s_2} > \eta_{s_1,s_2}$, so the audit remains risk-limiting. ◀

5.6 Simulations

This section simulates the suggested census RLA on the Cypriot census and its resulting allocation of representatives to districts in the House of Representatives of Cyprus. Our original intention was to simulate the suggested census RLA method on the US census and its resulting allocation of representatives in the US House of Representatives to the states. This turned out to be infeasible, however, as the audit outputted an insufficient risk-limit. This is a result of the relatively large number of states (50) and representatives (435) in the American system. Allocating many representatives to many states increases the probability of there being a single representative whose allocation is determined by a very small number of state residents.

To show that the census RLA is useful in other settings, we chose to simulate the audit on the House of representatives of Cyprus, where 56 representatives are allocated to 5 districts. This should be viewed as a pet-setting for testing the census RLA method, and not as a ready-as-is implementation for the Cypriot system.

The House of Representatives of Cyprus

The House of representatives of Cyprus is its sole legislating body, and holds 56 occupied seats. An additional 24 seats are reserved for the Turkish Cypriot community, who withdrew from the political decision-making process in 1964, leaving their house seats vacant [4].

The remaining 56 seats of the house are allocated to 5 districts. Currently, the allocation of seats to the districts is amended by law when found necessary, and does not change automatically following a census according to a set method. A census RLA could be useful when performing these amendments, to ensure that the resulting allocation of seats to districts is sufficiently reliable.

Technical Details

We allocated representatives to districts using the D'Hondt method. D'Hondt was chosen since it's currently used in the Cypriot elections to allocate seats to political parties. The audit was run assuming that each household holds 15 residents at most, and with $\delta = 10^{-10}$.

The census data used in the simulation is based on the results of the 2021 Cypriot census. For more details regarding the census data generation, see Appendix C. The simulation's code is available at the paper's GitHub repository (see title page).

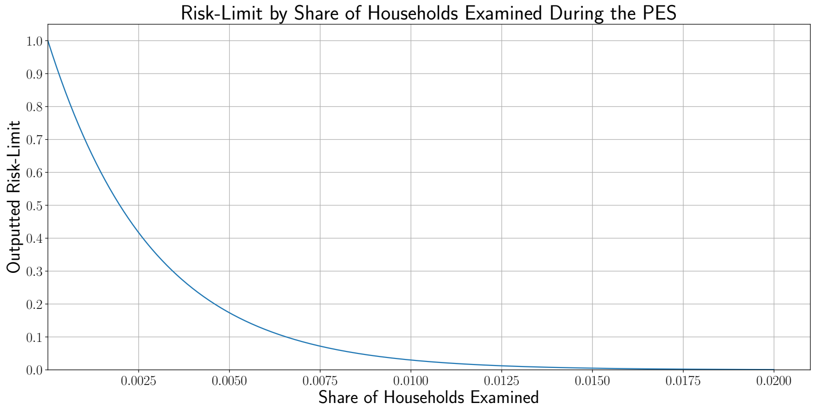

Figure 2 The census RLA output when the census and PES fully agree on the number of residents in each household, as a factor of the share of households that were surveyed by the PES.

Results

To examine the census RLA method, we present in Figure 2 the outputted risk-limit of the census RLA as a factor of the size of the PES. This simulation assumes that the census and the PES agree on the number of residents in each PES-surveyed households. Results with small census and PES disagreement, which are largely similar to the ones presented here, are available at the paper's GitHub repository.

Under the specified conditions, a PES which samples 0.66% of households is sufficient for a risk limit of 0.1, and a sample of 0.85% is sufficient for a risk-limit of 0.05. A PES often surveys around 1% of households [7], meaning that our census RLA can confidently approve the census' resulting allocation of representatives to districts under these conditions.

These results show that the census RLA method is applicable in some settings, when the number of representatives and federal-states to allocate them to is relatively small. When there are many representatives and federal-states, even a small error in the census can lead to a wrongful allocation of representatives, and auditing the census results requires a larger PES sample.

6 Discussion and Further Research

Throughout this work, we can observe that an election's social choice function and setting can severely limit the efficiency of their RLAs. Systems like the Israeli Knesset elections and the US House of Representatives' allocation of representatives to states are very sensitive to enumeration errors, making it difficult to audit them efficiently.

The simulation of the Batchcomp method on the Israeli Knesset elections (Section 4.3) indicates that Batchcomp provides a noticeable improvement over ALPHA-Batch in the limited settings that were tested. Despite this relative success, we cannot definitively say it outperforms existing methods without a clear, rigorous way of analyzing their efficiency.

The census RLA method appears to be useful in some limited settings, and can be implemented using existing post-enumeration surveys. In systems where our method is currently not sufficient, a census RLA could perhaps aim for a weaker guarantee - that the number of representatives each state should receive according to the PES is close to the number it has according to the census. This option is discussed in Appendix D.

The work raises many open questions and potential research directions:

Applying RLAs in Additional Settings: Generally speaking, RLAs can be applied whenever one wishes to verify the computation of some function over a large number of inputs obtained through potentially error-prone processes. While political elections provide a natural environment for their application, we advocate for their use in a wider range of settings to ensure reliable results.

As an example of such settings, RLAs could potentially be used to verify that decisions taken based on datasets which were altered in order to satisfy differential privacy are correct according to the real data. This could be achieved by running an RLA in a protected environment (enclave) which holds a subsample of the original, noiseless data. In this setting, the noisy, (differential private) dataset is seen as the reported result, while the noiseless dataset is the true results. An RLA can verify that the results of some computation over the differential private dataset and over the original noiseless dataset are likely to be identical, based on a (hopefully) small random sample from the original dataset. One challenge is to make sure that the very fact that the data passed the test does not hurt the desired differential privacy property.

Analytical Analysis of the Efficiency of RLAs: Most recent literature in the field, including this work, focuses on suggesting new RLA algorithms and fitting them to additional electoral systems and settings. There is little to no analytical analysis of the *efficiency* and capabilities of many RLA methods. Without a more rigorous analysis, it is not possible to definitively determine which RLA methods are better for which settings. Such analysis could help, for instance, to argue analytically whether Batchcomp is indeed preferable over ALPHA-Batch.

Connection Between RLAs and Computational Models: Thus far, advancements in the field of RLAs were done mostly independently and without connection to computational models. Finding such connections may inspire new RLA algorithms, or suggest new methods for analyzing the capabilities and efficiency of existing methods. As an example of these connections, RLAs can essentially be viewed as randomized decision trees, where each branch represents a different sequence of paper-ballots that can be uncovered during the audit. Viewing RLAs in this manner may allow us to analyze their query complexity (number of ballots examined) or instance complexity (best possible performance over specific election results) and to apply existing results from other fields onto RLAs.

As a potential example for this, viewing RLAs as randomized decision trees may allow us to find lower bounds for the query-complexity of RLAs by analyzing the randomized unlabeled certificate complexity of the social choice function they operate on, as defined by Grossman, Komargodski and Naor [6]. Randomized unlabeled certificate complexity is a complexity measure of a function over some specific input. It's defined roughly as the minimal number of queries, in expectation, that any randomized decision tree which computes this function has to perform over the specified input, given a permuted version of it as a certificate. This notion could be relevant for RLAs since they are essentially randomized decision trees which calculate a social choice function's output (the true winners) while using the reported election results as a certificate.

References

1 Michelle Blom, Jurlind Budurushi, Ronald L. Rivest, Philip B. Stark, Peter J. Stuckey, Vanessa Teague, and Damjan Vukcevic. Assertion-based approaches to auditing complex elections, with application to party-list proportional elections. In *International Joint Conference on Electronic Voting*, pages 47–62. Springer, 2021.

2 Michelle Blom, Philip B Stark, Peter J Stuckey, Vanessa Teague, and Damjan Vukcevic. Auditing hamiltonian elections. In *Financial Cryptography and Data Security. FC 2021 International Workshops: CoDecFin, DeFi, VOTING, and WTSC, Virtual Event, March 5, 2021, Revised Selected Papers*, pages 235–250. Springer, 2021.

3 United States Census Bureau. Historical households tables. `https://www.census.gov/data/tables/time-series/demo/families/households.html`, 2022. Table HH-4.

4 Giorgos Charalambous. The house of representatives. *The Politics and Government of Cyprus. Oxford: Peter Lange*, pages 143–168, 2008.

5 Michael Gallagher. Proportionality, disproportionality and electoral systems. *Electoral studies*, 10(1):33–51, 1991.

6 Tomer Grossman, Ilan Komargodski, and Moni Naor. Instance complexity and unlabeled certificates in the decision tree model. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

7 Guihua Hu, Ting Wen, and Yuhuan Liu. Determining the sample size of a post-enumeration survey: The case of china, 2020. *Mathematical Population Studies*, pages 1–31, 2022.

8 The Knesset Lexicon. The distribution of knesset seats (bader-ofer method). `https://m.knesset.gov.il/en/about/lexicon/pages/seats.aspx`. This page wrongly states that the electoral threshold is 2.0%. It was since changed to 3.25%.

9 Mark Lindeman and Philip B. Stark. A gentle introduction to risk-limiting audits. *IEEE Security & Privacy*, 10(5):42–49, 2012.

10 Statistical Service of the Republic of Cyprus. Census of population and housing 2021: Preliminary results. `https://www.pio.gov.cy/en/press-releases-article.html?id=27965`, 2022.

11 Philip B. Stark. Sets of half-average nulls generate risk-limiting audits: Shangrla. In *International Conference on Financial Cryptography and Data Security*, pages 319–336. Springer, 2020.

12 Philip B. Stark. Alpha: Audit that learns from previously hand-audited ballots. *The Annals of Applied Statistics*, 17(1):641–679, 2023.

13 Department of Economics United Nations Secretariat and Statistics Division Social Affairs. Post enumeration surveys operational guidelines. `https://unstats.un.org/unsd/demographic/standmeth/handbooks/Manual_PESen.pdf`, 2010.

14 Ian Waudby-Smith, Philip B. Stark, and Aaditya Ramdas. Rilacs: Risk limiting audits via confidence sequences. In *International Joint Conference on Electronic Voting*, pages 124–139. Springer, 2021.

A Proofs

A.1 Proof of Theorem 14

▶ **Theorem 14.** *Assume the PES surveyed all households. The allocation of representatives according to the census and according to the PES match, iff for any two states $s_1, s_2 \in \mathcal{S}$:*

$$\frac{\sum_{h\in H} g_{s_1}^{PES}(h) + c_{s_1}}{d(r^{cen}(s_1))} > \frac{\sum_{h\in H} g_{s_2}^{PES}(h) + c_{s_2}}{d(r^{cen}(s_2)+1)}. \tag{7}$$

Proof. First, assume that the two allocations of representatives match. Examine the imaginary table with which representatives are allocated to states according to the PES. Recall that each state has exactly its first $r^{PES}(s)$ cells marked. Since we assume that for any $s \in \mathcal{S}$, $r^{PES}(s) = r^{cen}(s)$, we have it that for any $s_1, s_2 \in \mathcal{S}$, the cell at index $[s_1, r^{cen}(s_1)]$ is marked, while the cell at $[s_2, r^{cen}(s_2)+1]$ is not. Since the marked cells are the ones which hold the largest values in the table, the cell at $[s_1, r^{cen}(s_1)]$ has a larger value than the cell at $[s_2, r^{cen}(s_2)+1]$. Writing these values out results exactly in (7)- the larger term is the value at $[s_2, r^{cen}(s_2)+1]$, and the smaller is the value at $[s_1, r^{cen}(s_1)]$.

Towards proving the other direction of the equivalence, we show that if (7) is true for any $s_1, s_2 \in \mathcal{S}$, then a certain condition (9) holds for any s_1, s_2. We then show that if this condition is true, then the allocation of representatives according to the census and according to the PES match.

▷ Claim 21. Let $r^{PES}(s)$ be the number of representatives a state s is allocated according to the full PES results. For any $s_1, s_2 \in \mathcal{S}$, if (7) is true then:

$$\left(r^{PES}(s_1) \geq r^{cen}(s_1)\right) \vee \left(r^{PES}(s_2) \leq r^{cen}(s_2)\right). \tag{9}$$

Proof. Assume towards contradiction that for some $s_1, s_2 \in \mathcal{S}$, the condition (9) is false, meaning that $\left(r^{PES}(s_1) < r^{cen}(s_1)\right) \wedge \left(r^{PES}(s_2) > r^{cen}(s_2)\right)$ is true.

Examine the table used to allocate representatives to states according to the PES results. According to this table, s_2 is awarded $r^{PES}(s_2)$ representatives. Since $r^{PES}(s_2) > r^{cen}(s_2)$, and since the row s_2 has exactly its first $r^{PES}(s_2)$ cells marked, the cell at $[s_2, r^{cen}(s_2)+1]$ is marked. Additionally, since s_1 was awarded exactly $r^{PES}(s_1)$ seats and since $r^{PES}(s_1) < r^{cen}(s_1)$, the cell at $[s_1, r^{cen}(s_1)]$ is not marked.

By the paragraph above, if $\left(r^{PES}(s_1) \geq r^{cen}(s_1)\right) \vee \left(r^{PES}(s_2) \leq r^{cen}(s_2)\right)$ is false, then the cell at $[s_2, r^{cen}(s_2)+1]$ is marked while the cell at $[s_1, r^{cen}(s_1)]$ is not. Since the marked cells are the ones which hold the largest values, it follows that the cell at $[s_2, r^{cen}(s_2)+1]$ has a larger value than the cell at $[s_1, r^{cen}(s_1)]$, meaning that:

$$\frac{\sum_{h\in H} g_{s_1}^{PES}(h) + c_{s_1}}{d(r^{cen}(s_1))} \leq \frac{\sum_{h\in H} g_{s_2}^{PES}(h) + c_{s_2}}{d(r^{cen}(s_2)+1)}.$$

The larger term in this inequality is the value at index $[s_2, r^{cen}(s_2)+1]$ and the smaller one is the value at index $[s_1, r^{cen}(s_1)]$. This contradicts (7), and thereby proves this claim. ◁

▷ Claim 22. If (9) is true for any $s_1, s_2 \in \mathcal{S}$, then the allocation of representatives according to the census and according to the full PES are identical.

Proof. Assume towards contradiction that the two allocations are not identical. Therefore, there must be at least one state s with $r^{PES}(s) \neq r^{cen}(s)$. If $r^{PES}(s) > r^{cen}(s)$, since the number of total representatives is constant, there must be another state s' with $r^{PES}(s') < r^{cen}(s')$. Similarly, if $r^{PES}(s) < r^{cen}(s)$, there must be another state s' with $r^{PES}(s') > r^{cen}(s')$. Either way, (9) is false. Thus, if (9) is true for every pair of states, then the two allocations must be identical, completing the proof. ◁

Using these two claims, we can now complete the proof of this theorem. Assume (7) is true for any pair of states. By Claim 21, (9) is also true for any pair of states, and by Claim 22, this makes the allocation of representatives according to the census and according to the PES identical. This proves the other direction of the equivalence and concludes the proof. ◀

A.2 Proof of Theorem 19

▶ **Theorem 19.** *Assume that the PES surveyed all households. The assorters $\{A_{s_1,s_2} \mid s_1, s_2 \in \mathcal{S}\}$, as defined in Definition 18, are all non-negative and satisfy the following condition: the allocation of representatives according to the census and the PES match iff for all $s_1, s_2 \in \mathcal{S}$:*

$$\frac{1}{|H|}\sum_{h\in H} A_{s_1,s_2}(h) > \frac{1}{2}.$$

Proof. We show that A_{s_1,s_2} is non-negative and that the required equivalence holds.

▷ Claim 23. For any $s_1, s_2 \in \mathcal{S}$, A_{s_1,s_2} is non-negative.

Proof. Fix two states $s_1, s_2 \in \mathcal{S}$. Recall the definition of A_{s_1,s_2}:

$$A_{s_1,s_2}(h) := \frac{1}{2} + \frac{m_{s_1,s_2} + a^{PES}_{s_1,s_2}(h) - a^{cen}_{s_1,s_2}(h)}{2(z_{s_1,s_2} - m_{s_1,s_2})}.$$

By the definition of $a^{PES}_{s_1,s_2}$ and $a^{cen}_{s_1,s_2}$, the value of the nominator in the expression above is:

$$m_{s_1,s_2} + a^{PES}_{s_1,s_2}(h) - a^{cen}_{s_1,s_2}(h) = m_{s_1,s_2} + \frac{g^{PES}_{s_1}(h) - g^{cen}_{s_1}(h)}{cd(r^{cen}(s_1))} + \frac{g^{cen}_{s_2}(h) - g^{PES}_{s_2}(h)}{cd(r^{cen}(s_2)+1)}.$$

h is either from s_1, from s_2 or from neither of them. If it's from neither, this expression equals m_{s_1,s_2}. If it's from s_1, then:

$$m_{s_1,s_2} + \frac{\overbrace{g^{PES}_{s_1}(h)}^{\geq 0} - \overbrace{g^{cen}_{s_1}(h)}^{\leq g^{max}}}{cd(r^{cen}(s_1))} + \frac{\overbrace{g^{cen}_{s_2}(h) - g^{PES}_{s_2}(h)}^{=0}}{cd(r^{cen}(s_2)+1)} \geq m_{s_1,s_2} - \frac{g^{max}}{cd(r^{cen}(s_1))}$$

where g^{max} is the maximal number of residents a single household may have. If h is from s_2, then:

$$m_{s_1,s_2} + \frac{\overbrace{g^{PES}_{s_1}(h) - g^{cen}_{s_1}(h)}^{=0}}{cd(r^{cen}(s_1))} + \frac{\overbrace{g^{cen}_{s_2}(h)}^{\geq 0} - \overbrace{g^{PES}_{s_2}(h)}^{\leq g^{max}}}{cd(r^{cen}(s_2)+1)} \geq m_{s_1,s_2} - \frac{g^{max}}{cd(r^{cen}(s_2)+1)}.$$

So for any $h \in H$:

$$\begin{aligned} & m_{s_1,s_2} + a^{PES}_{s_1,s_2}(h) - a^{cen}_{s_1,s_2}(h) \\ & \geq \min\left\{ m_{s_1,s_2}, m_{s_1,s_2} - \frac{g^{max}}{cd(r^{cen}(s_2)+1)}, m_{s_1,s_2} - \frac{g^{max}}{cd(r^{cen}(s_1))} \right\}. \end{aligned} \tag{10}$$

By (10) and by the definition of z_{s_1,s_2} (Definition 18):

$$m_{s_1,s_2} + a^{PES}_{s_1,s_2}(h) - a^{cen}_{s_1,s_2}(h) \geq m_{s_1,s_2} - z_{s_1,s_2}. \tag{11}$$

Meaning that for any $h \in H$:

$$A_{s_1,s_2}(h) = \frac{1}{2} + \frac{m_{s_1,s_2} + a^{PES}_{s_1,s_2}(h) - a^{cen}_{s_1,s_2}(h)}{2(z_{s_1,s_2} - m_{s_1,s_2})} \geq \frac{1}{2} + \frac{m_{s_1,s_2} - z_{s_1,s_2}}{2(z_{s_1,s_2} - m_{s_1,s_2})} = 0,$$

proving the claim. ◁

▷ Claim 24. Assume the PES surveyed all households. The allocation of representatives according to the census and the PES match iff for all $s_1, s_2 \in \mathcal{S}$:

$$\frac{1}{|H|} \sum_{h \in H} A_{s_1,s_2}(h) > \frac{1}{2}.$$

Proof. By Theorem 16, the allocation of representatives according to the census and the PES match iff for all $s_1, s_2 \in \mathcal{S}$:

$$\frac{1}{|H|} \sum_{h \in H} a^{PES}_{s_1,s_2}(h) > \frac{1}{2}.$$

Therefore, to prove this claim, it suffices to prove that for every $s_1, s_2 \in \mathcal{S}$:

$$\left(\frac{1}{|H|}\sum_{h\in H} A_{s_1,s_2}(h) > \frac{1}{2}\right) \iff \left(\frac{1}{|H|}\sum_{h\in H} a^{PES}_{s_1,s_2}(h) > \frac{1}{2}\right).$$

Fix any two federal-states $s_1, s_2 \in \mathcal{S}$. We show that the two inequalities above are equivalent:

$$\begin{aligned} & \frac{1}{|H|}\sum_{h\in H} A_{s_1,s_2}(h) > \frac{1}{2} \\ \iff & \frac{1}{|H|}\sum_{h\in H}\left(\frac{1}{2} + \frac{m_{s_1,s_2} + a^{PES}_{s_1,s_2}(h) - a^{cen}_{s_1,s_2}(h)}{2(z_{s_1,s_2} - m_{s_1,s_2})}\right) > \frac{1}{2} \\ \iff & \frac{1}{|H|}\sum_{h\in H}\frac{m_{s_1,s_2} + a^{PES}_{s_1,s_2}(h) - a^{cen}_{s_1,s_2}(h)}{2(z_{s_1,s_2} - m_{s_1,s_2})} > 0. \end{aligned}$$

Now, using the definition of m_{s_1,s_2} and re-arranging the summation yields the desired equivalence:

$$\begin{aligned} \iff & \frac{1}{|H|}\sum_{h\in H}\frac{\frac{1}{|H|}\sum_{h'\in H} a^{cen}_{s_1,s_2}(h') - \frac{1}{2} + a^{PES}_{s_1,s_2}(h) - a^{cen}_{s_1,s_2}(h)}{2(z_{s_1,s_2} - m_{s_1,s_2})} > 0 \\ \iff & \frac{\frac{1}{|H|}\sum_{h\in H} a^{PES}_{s_1,s_2}(h) - \frac{1}{2}}{2(z_{s_1,s_2} - m_{s_1,s_2})} > 0 \\ \iff & \frac{1}{|H|}\sum_{h\in H} a^{PES}_{s_1,s_2}(h) > \frac{1}{2}. \end{aligned}$$

The last transition relies on the fact that $z_{s_1,s_2} > m_{s_1,s_2}$, which is true since $z_{s_1,s_2} \geq \max_{h\in H} a^{cen}(h) \geq m_{s_1,s_2}$ (see Definition 18). This concludes the proof of this claim. ◁

The combination of these two claims completes this theorem's proof. ◀

B Batchcomp – Choosing δ

As seen in Section 3.5, for every assorter a_k and its Batchcomp counterpart A_k we initialize:

$$U_k = \frac{1}{2} + \frac{M_k + \delta}{2(w_k - M_k)},$$

where $\delta > 0$. Different choices for δ all maintain the RLA guarantee, but under certain conditions, certain values of δ yield more efficient audits. This section attempts to give intuition regarding the ideal choice of δ. Generally, the more we expect the reported vote tallies of the different batches to be accurate, the smaller δ should be. We show this by comparing μ_k to the expected value of a Batchcomp assorter on the next batch to be sampled.

▷ Claim 25. During a Batchcomp RLA, if the next sampled batch B_i satisfies $A_k(B_i) \geq \mu_k$ for some batch-assorter A_k, then choosing a smaller U_k increases the audit's efficiency, and vice-versa; if $A_k(B_i) < \mu_k$, then setting a larger U_k increases the audit's efficiency.

Proof. Examine some Batchcomp assorter A_k. Approving its assertion requires fewer ballots the more significantly T_k grows per batch. This is because the audit approves assertion k when $T_k > \frac{1}{\alpha}$. Therefore, it suffices to show that if $A_k(B_i) \geq \mu_k$, then T_k grows more significantly when U_k is small, and vice-versa.

Towards this purpose, denote the next audited batch as B_i. To prove this claim, we take the derivative by U_k of the update rule of T_k in step 2c of the Batchcomp algorithm:

$$T_k \leftarrow T_k \left(\frac{A_k(B_i)}{\mu_k} \frac{\eta_k - \mu_k}{U_k - \mu_k} + \frac{U_k - \eta_k}{U_k - \mu_k} \right).$$

Taking its derivative by U_k results in:

$$\begin{aligned}
&T_k \left(-\frac{A_k(B_i)}{\mu_k} \frac{\eta_k - \mu_k}{(U_k - \mu_k)^2} + \frac{1}{U_k - \mu_k} - \frac{U_k - \eta_k}{(U_k - \mu_k)^2} \right) \\
=&\frac{T_k}{(U_k - \mu_k)^2} \left(-\frac{A_k(B_i)}{\mu_k} (\eta_k - \mu_k) + U_k - \mu_k - U_k + \eta_k \right) \\
=&\frac{T_k}{(U_k - \mu_k)^2} \left(-\frac{A_k(B_i)}{\mu_k} (\eta_k - \mu_k) - \mu_k + \eta_k \right) \\
=&\underbrace{T_k \frac{\eta_k - \mu_k}{(U_k - \mu_k)^2}}_{>0} \left(1 - \frac{A_k(B_i)}{\mu_k} \right).
\end{aligned}$$

Where the term above the under-brace is positive since T_k is positive, and since we always have $U_k > \eta_k > \mu_k \geq 0$. We can observe that if $A_k(B_i) > \mu_k$, this derivative is negative, meaning that choosing a smaller value for U_k causes T_k to increase more significantly. If $A_k(B_i) < \mu_k$, then the opposite is true. This concludes the proof of this claim. $\triangleleft$

According to this claim, if we expect to have $A_k(B_i) > \mu_k$ for all batch-assorters and batches, we should choose a smaller δ, and vice versa. When using a Batchcomp assorter, we have:

$$A_k(B_i) = \frac{1}{2} + \frac{M_k + a_k^{true}(B_i) - a_k^{rep}(B_i)}{2(w_k - M_k)}.$$

And $w_k > M_k > 0$ by the definition of M_k. Therefore, as long as the batch-level discrepancies between the reported and true vote counts are small, we expect to consistently have $A_k(B_i) \geq \mu_k$, meaning we should choose a smaller δ. To get $A_k(B_i) < \mu_k$, we would need to have $a_k^{true}(B_i) - a_k^{rep}(B_i) > M_k$, meaning that the discrepancy in vote counts, as it relates to the assorter a_k, is greater than its reported margin. If the margin isn't extremely small, and the errors in the vote count are uncorrelated and rare, this is very unlikely to happen. We believe that this should encourage choosing a very small value for δ, since it would only make the audit inefficient if it's likely that the vote counting was malicious.

▶ **Conclusion** (informal). *A Batchcomp RLA is more efficient when $\delta > 0$ is very small, as long as the vote tallying is not done maliciously.*

C Census RLA – Data Generation

The data used to perform this simulation is based on the population census conducted in 2021 [10]. The Statistical Service of Cyprus publicly reports the total number of residents in every district, but not the individual household data, which the census RLA requires. To generate this data, we assumed that the number of residents per household distributes as it does in the United States, as reported by its census [3]. We additionally assumed that 1% of households do not respond to the census and are counted as if they have no residents. The per-household data used in these simulations was generated as follows:

1. The number of households per district was calculated by dividing the district's population by the expected number of residents per household.

2. The number of residents in each household was drawn from the distribution specified in the US census [3].
3. Due to the randomness involved in the previous step, the real census and our generated one might disagree on the population of the districts. To balance this, the constant of each district (c_s in (5) at Section 5.2) was set as the difference between the population of the district according to the real census and according to our generated one. With this definition, the allocations of representatives to districts by the real census and by our generated one are necessarily identical.

D Weakening the RLA Guarantee

When conducting a SHANGRLA based RLA, a single assertion may be the difference between reading relatively few or a relatively many ballots or households to approve the reported outcome. As an example of this, in the Knesset elections examined in Section 4.3, a single assertion causes the Batchcomp audit to read 85% of ballots, instead of only 32% without it. In such cases, the auditing body may decide in advance that a certain assertion is too difficult to audit, and forgo approving it. This decision can be taken based on the assertion's margin, or by simulating the audit in advance and checking the number of ballots required per assertion.

For RLAs which approve an allocation of parliament seats to different political parties or federal-states, tight assertions can be altered to verify that the reported allocation of seats is nearly accurate. For example, in the Knesset elections, if an assertion which involves some specific party drastically increases the number of ballots the audit reads, an RLA can approve that the number of seats that this party wins according to the reported results is at most ± 1 from its true number. This would result in a shorter audit, at the expense of a slightly weaker guarantee.

To achieve this, when designing move-seat assertions which involve some difficult-to-audit party p, we alter the number of seats it reportedly won. For every assertion which verifies that seats shouldn't be moved *from* p to some other party p', we imagine p reportedly won one seat less than it actually did. Similarly, when verifying that seats shouldn't be moved from p' *to* p, we imagine p has reportedly won one seat more than it did. The same notion also applies for census RLAs and their assorters.

Alternatively, if some assertions are too difficult to audit, the auditing body can decide to verify that certain blocks of parties or federal-states get the correct number of seats. For parliamentary elections, this is achieved by partitioning the parties to electoral blocks, and verifying that no seats should be moved between every two parties who belong to different blocks.