Risk-Limiting Audits for IRV Elections

Michelle Blom*, Peter J. Stuckey† and Vanessa Teague‡

March 22, 2019

Abstract

Risk-limiting post election audits guarantee a high probability of correcting incorrect election results, independent of why the result was incorrect. Ballot-polling audits select ballots at random and interpret those ballots as evidence for and against the reported result, continuing this process until either they support the recorded result, or they fall back to a full manual recount. For elections with digitised scanning and counting of ballots, a comparison audit compares randomly selected digital ballots with their paper versions. Discrepancies are referred to as errors, and are used to build evidence against or in support of the recorded result. Risk-limiting audits for first-past-the-post elections are well understood, and used in some US elections. We define a number of approaches to ballot-polling and comparison risk-limiting audits for Instant Runoff Voting (IRV) elections. We show that for almost all real elections we found, we can perform a risk-limiting audit by looking at only a small fraction of the total ballots (assuming no errors were made in the tallying and distribution of votes).

1 Introduction

Instant Runoff Voting (IRV) is a system of preferential voting in which voters rank candidates in order of preference. IRV is used for all parliamentary lower house elections in Australia, parliamentary elections in Fiji and Papua New Guinea, presidential elections in Ireland and Bosnia/Herzogovinia, and local elections in numerous locations world-wide, including the UK and United States. Given candidates c_1, c_2, c_3, and c_4, each vote in an IRV election is a (*possibly partial*) ranking of these candidates. A vote with the ranking $[c_1, c_2, c_3]$ expresses a first preference for candidate c_1, a second preference for c_2, and a third for

*michelle.blom@unimelb.edu.au
The University of Melbourne
†peter.stuckey@monash.edu
Monash University
‡michelle.blom@unimelb.edu.au
The University of Melbourne

c_3. The tallying of votes proceeds by distributing each vote to its first ranked candidate. The candidate with the fewest votes is eliminated, with their votes redistributed to subsequent, less preferred candidates. Elimination proceeds in this fashion, until a single candidate w remains, who is declared the winner.

The scanning and digitisation of ballots, and the use of automated counting software for computing the outcomes of elections, is becoming more commonplace. In light of recent attempts by foreign powers to interfere in electoral processes in the US [12], there is a growing need for efficient and statistically sound electoral audits. *Risk Limiting Audits* [7] (RLAs) provide strong statistical evidence that the reported outcome of an election is correct, or revert to a manual recount if it is wrong. The probability that the audit fails to detect a wrong outcome is bounded by a *risk limit*. An RLA with a risk limit of 1%, for example, has at most a 1% chance of failing to detect that a reported election outcome is wrong. In this paper we present several methods for undertaking both ballot-polling and ballot-level comparison RLAs of IRV elections, and compare the auditing effort required by each on a set of real IRV election instances. We show, in this paper, that we can design risk limiting audits for IRV elections that, in general, require only a small fraction of cast ballots to be sampled.

Risk limiting ballot-polling and ballot-level comparison audits have been developed for first-past-the-post (FPTP) or k-winner plurality elections [8, 19]. In such elections, the k candidates with the most votes are declared winners. A ballot-polling RLA of such an election randomly samples the paper ballots cast (or records produced). As each ballot is examined, we update a series of statistics representing our hypotheses that any loser actually received more votes than any winner. Once we have seen enough ballots to be confident that all these hypotheses can be rejected, the reported outcome is correct and the audit concludes. At any point, the audit can fall back to a full manual recount, for example if it is taking too long or has examined a large number of ballots. It is designed so that the probability of concluding with acceptance, when the result is in fact wrong, is at most α. Ballot-level comparison RLAs are applicable in settings where paper ballots have been scanned and digitised, or a paper-based electronic voting system has been used, producing an index that allows individual electronic ballots to be matched to the paper ballot they represent. Each sampled ballot is compared to its corresponding electronic record. An erroneous ballot is one that does not match its electronic record. These errors are then used to update a similar set of statistics representing our hypotheses that the reported election outcomes are actually wrong.

In this paper we present several methods for undertaking ballot-polling RLAs of IRV elections, by adapting a ballot-polling RLA method (BRAVO) designed for first-past-the-post (FPTP) or k-winner plurality elections [8]. The most straightforward of these methods views, and audits, each round of an IRV election as a multiple-winner plurality election. A more efficient method, requiring fewer ballot samples, seeks to prove that the reported winner could not have been eliminated before any other candidate. The former approach is designed to confirm the correctness of each elimination in the IRV counting process, while the latter aims to to confirm *only* that the reported winner of the election is correct. We also adapt a comparison-based RLA method [19], similarly designed for FPTP elections, to IRV. We describe, and evaluate, each of the auditing methods previously discussed in the comparison audit context. We compare the relative efficiency of ballot-polling and comparison-based RLAs, in terms of the level of auditing effort required, on a suite of IRV elections. As for FPTP elections, comparison RLAs require fewer ballot samples, in general, to confirm the correctness of an election result. This is because they can assess the differences between reported and actual individual ballots, a significant source of extra information that becomes particularly advantageous when the margin is small.

We present an algorithm, *audit-irv*, for generating a collection of *facts* to audit for a given IRV election that, if confirmed with a given degree of statistical confidence, confirms that the reported winner of the election is the correct winner to the given degree of statistical confidence. The *audit-irv* algorithm finds the set of such facts that require the least estimated level of auditing effort to prove. We apply this algorithm to generate efficient ballot-polling and comparison RLAs for IRV elections. We experimentally consider audits where all votes are indeed recorded correctly, and elections where discrepancies exist between paper ballots and their electronic records.

The contributions of this paper are:

- The first use of ballot-polling for IRV elections beyond the simplest approach of treating each round in the IRV election as a FPTP election previously described (but not evaluated) by Sarwate *et al* [14].
- The first use of ballot-level comparison auditing for IRV elections.
- An efficient algorithm for determining a small set of facts to be verified by either a ballot-polling or ballot-level comparison audit in order to be confident within some given risk limit that the winner of an IRV election is the correct winner.

This paper is structured as follows. Related work is described in Section 2. Required background and definitions are presented in Section 3. Section 4 describes the ballot-polling [8] and ballot-level comparison-based RLAs [19] upon which our IRV audits are based. We present our IRV ballot-polling and comparison-based RLAs in Sections 5 and 6, and evaluate their efficiency in Section 7.

2 Related Work

Post-election audits are a key measure for increasing both security in our electoral systems, and public confidence in the integrity of our elections [12]. Risk-limiting audits of reported election results against paper ballots (or records) represent the current best-practice for verifying the integrity of an election [13], and a central component of conducting evidence-based elections [9].

There is a growing literature on the use of risk-limiting audits for auditing the outcome of varying types of election (see e.g. [15, 6, 5, 3, 8, 16, 14]). Risk-limiting audits (RLAs) have been applied to a number of plurality (first-past-the-post) elections, including four 2008 elections in California [6] and elections in over 50 Colorado counties in 2017 [10]. Stark and Teague [20] present RLAs for D'Hondt (and similar) elections, applicable in a number of European countries such as Norway, Germany, Luxembourg, and Denmark. General auditing procedures designed to enhance electoral integrity have been outlined by Antoyan *et al* [1]. The BRAVO ballot-polling risk-limiting audit [8], designed for FPTP elections, forms the basis of our IRV ballot-polling audits. The ballot-level comparison RLA presented by Stark [19] forms the basis of our IRV comparison RLAs.

A straightforward RLA of an IRV election can be conducted by treating each IRV round as a separate FPTP election. This idea was described by Sarwate *et al* [14] although not explored in any detail. Sarwate *et al* [14] consider two additional approaches for designing a comparison audit of an IRV election. The first of these methods determines whether replacing an erroneous ballot with its correct representation changes the margin of victory of the election. The second samples K ballots and checks whether the number of erroneous ballots in the sample exceeds a threshold whose value is based on the election's margin of victory. We demonstrate, however, that we can more efficiently audit an IRV election outcome by simply verifying that the reported winner was not defeated by any other candidate.

The margin of victory of the election provides an indication of how many ballots will need to be sampled during a RLA. Automatic methods for computing electoral margins for IRV elections have been presented by a number of authors [11, 4, 2].

3 Preliminaries

In a first-past-the-post (FPTP) election, a voter marks a single candidate on their ballot when casting their vote. The candidate who receives the most votes is declared the winner. The BRAVO ballot-polling risk limiting audits [8], and the MACRO ballot-level comparison risk-limiting audits [19], are designed for k-winner FPTP contests. A voter may vote for up to k of the candidates on their ballot, and the k candidates with the highest number of votes are declared winners. IRV, in contrast, is a form of preferential voting in which voters express a preference ordering over a set of candidates on their ballot. The tallying of votes in an IRV election proceeds by a series of rounds in which the candidate with the fewest votes is eliminated (see Figure 1) with the last remaining candidate declared the winner. All ballots in an eliminated candidate's tally are distributed to the next most-preferred (remaining) candidate in their ranking.

Let $\mathcal{C}$ be the set of candidates in an IRV election $\mathcal{B}$. We refer to sequences of candidates π in list notation (e.g., $\pi = [c_1, c_2, c_3, c_4]$), and use such sequences to represent both votes and elimination orders. An election $\mathcal{B}$ is defined as

Initially, all candidates remain standing (are not eliminated)
While there is *more than one* candidate standing
  For every candidate c standing
    Tally (count) the ballots in which c is the highest-ranked
    candidate of those standing
  Eliminate the candidate with the smallest tally
The winner is the one candidate not eliminated

Figure 1: An informal definition of the IRV (also known as Alternate Vote) counting algorithm.

a multiset[1] of ballots, each ballot $b \in \mathcal{B}$ a sequence of candidates in $\mathcal{C}$, with no duplicates, listed in order of preference (most preferred to least preferred). Throughout this paper we use the notation $\mathit{first}(\pi) = \pi(1)$ to denote the first candidate in a sequence π. In each round of vote counting, there are a current set of eliminated candidates $\mathcal{E}$ and a current set of candidates still standing $\mathcal{S} = \mathcal{C} \setminus \mathcal{E}$. The winner c_w is the last standing candidate.

Definition 1. Projection $\mathbf{p}_{\mathcal{S}}(\pi)$ We define the projection of a sequence π onto a set $\mathcal{S}$ as the largest subsequence of π that contains only elements of $\mathcal{S}$. (The elements keep their relative order in π). For example:
$p_{\{c_2,c_3\}}([c_1, c_2, c_4, c_3]) = [c_2, c_3]$ and $p_{\{c_2,c_3,c_4,c_5\}}([c_6, c_4, c_7, c_2, c_1]) = [c_4, c_2]$.

Each candidate $c \in \mathcal{C}$ has a *tally* of ballots. Ballots are added to this tally upon the elimination of a candidate $c' \in \mathcal{C} \setminus c$, and are redistributed upon the elimination of c.

Definition 2. Tally $\mathbf{t}_{\mathcal{S}}(\mathbf{c})$ Given candidates $\mathcal{S} \subseteq \mathcal{C}$ are still standing in an election $\mathcal{B}$, the tally for a candidate $c \in \mathcal{C}$, denoted $t_{\mathcal{S}}(c)$, is defined as the number of ballots $b \in \mathcal{B}$ for which c is the most-preferred candidate of those remaining. Recall that $p_{\mathcal{S}}(b)$ denotes the sequence of candidates mentioned in b that are also in $\mathcal{S}$.

$$t_{\mathcal{S}}(c) = \ | \ [b \mid b \in \mathcal{B}, c = \mathit{first}(p_{\mathcal{S}}(b))] \ | \tag{1}$$

The *primary vote* of candidate $c \in \mathcal{C}$, denoted $f(c)$, is the number of ballots $b \in \mathcal{B}$ for which c is ranked highest. Note that $f(c) = t_{\mathcal{C}}(c)$.

$$f(c) = \ | \ [b \mid b \in \mathcal{B}, c = \mathit{first}(b)] \ | \tag{2}$$

Example 1. Consider the IRV election of Table 1. The tallies of c_1, c_2, c_3, and c_4, in the 1^{st} counting round are 26000, 10000, 9000, and 15000 votes. Candidate c_3 is eliminated, and 9000 ballots are distributed to c_4, who now has

[1] A multiset allows for the inclusion of duplicate items.

Ranking	Count
$[c_2, c_3]$	4000
$[c_1]$	20000
$[c_3, c_4]$	9000
$[c_2, c_3, c_4]$	6000
$[c_4, c_1, c_2]$	15000
$[c_1, c_3]$	6000

(a)

Candidate	Rnd1	Rnd2	Rnd3
c_1	26000	26000	26000
c_2	10000	10000	—
c_3	9000	—	—
c_4	15000	24000	30000
	$t_{\{c_1,c_2,c_3,c_4\}}$	$t_{\{c_1,c_2,c_4\}}$	$t_{\{c_1,c_4\}}$

(b)

Table 1: An example IRV election, stating (a) the number of ballots cast with each listed ranking over four candidates, and (b) the tallies after each round of counting.

a tally of 24000. Candidate c_2, on 10000 votes, is eliminated next with 6000 of their ballots given to c_4 (the remainder have no subsequent preferences and are exhausted). Candidates c_1 and c_4 remain with tallies of 26000 and 30000. Candidate c_1 is eliminated and c_4 elected. $\square$

4 Risk-limiting audits for FPTP

The aim of a risk limiting audit is to either gain evidence that the reported results are correct (to some risk limit α) or to correct an incorrect result by falling back to a manual recount. To this end we will consider two versions of the statistics defined in the previous section. We use the regular definition for the *recorded* values made during the election, and add a tilde $\tilde{}$ to mean the *actual* values which should have been calculated, as represented in the paper record. Hence $f(c)$ is the recorded primary vote for candidate c and $\tilde{f}(c)$ is the actual primary vote for the candidate.

For now we consider a simple k-winner from n candidates FPTP election where the k candidates who have the greatest number of votes are elected. All winners are elected simultaneously and there is no transfer of votes. Given a set of $\mathcal{C}$ candidates ($|\mathcal{C}| = n$) there will be a set of $\mathcal{W}$ *winners* ($|\mathcal{W}| = k$) and $\mathcal{L}$ *losers* ($|\mathcal{L}| = n - k$).

We now present the BRAVO algorithm [8] for ballot-polling risk-limiting audits of such elections (Figure 2) and a similar algorithm for conducting a risk-limiting ballot-level comparison audit (Figure 3), adapted from the MACRO algorithm [19]. Both methods are applicable in elections where each ballot may express a vote for one or more candidates. For our proposed IRV audits, we apply BRAVO and the ballot-level comparison RLA in contexts where each ballot represents a vote for a single candidate only (*i.e.*, in any round of an IRV count, each ballot belongs to the tally of no more than one candidate). We describe the BRAVO and ballot-level comparison RLA algorithms in the context where each ballot b is equivalent to *first*(b). Then $f(c)$ is the tally of votes for each candidate $c \in \mathcal{C}$.

```
bravo(B̃,𝒲,ℒ,α,M)
    for(w ∈ 𝒲, l ∈ ℒ)
        T_wl := 1
        s_wl := f(w)/(f(w) + f(l))
    H := 𝒲 × ℒ
    m := 0
    while(m < M ∧ H ≠ ∅)
        randomly draw ballot b from B̃
        m := m + 1
        if(first(b) ∈ 𝒲)
            for((w, l) ∈ H, w = first(b))
                T_wl := T_wl × 2s_wl
                if(T_wl ≥ 1/α)
                    % reject the null hypothesis
                    H = H − {(w, l)}
        else if(first(b) ∈ ℒ)
            for((w, l) ∈ H, l = first(b))
                T_wl := T_wl × 2(1 − s_wl)
    if(H = ∅)
        % reported results stand
        return true
    else % full recount required
        return false
```

Figure 2: BRAVO algorithm for a ballot-polling RLA audit of a FPTP election with actual ballots $\tilde{\mathcal{B}}$, declared winners $\mathcal{W}$, declared losers $\mathcal{L}$, risk limit α and limit on ballots checked M.

4.1 BRAVO: Ballot-polling Risk-Limiting Audits

A BRAVO audit independently tests $k(n-k)$ null hypotheses $\{\tilde{f}(w) \leq \tilde{f}(l)\}$ for each winner/loser pair, representing the hypothesis that l actually beat w. A statistic for each test $\{T_{wl}\}$ is updated when a ballot is drawn for either its winner or its loser.

Given an overall risk limit α we can estimate for each hypothesis the number of sampled ballots we expect will be required to reject the hypothesis assuming the announced election counts are perfectly accurate. Let p_c be the proportion of recorded votes for candidate c, i.e. $p_c = f(c)/|\mathcal{B}|$. Let s_{wl} be the proportion of recorded votes for the winner w of the votes for the winner and loser, $s_{wl} = p_w/(p_w + p_l)$. Clearly $s_{wl} > 0.5$. Then the *Average Sample Number (ASN)* for BRAVO, that is the expected number of samples to reject the null hypothesis $\{\tilde{p}_w \leq \tilde{p}_l\}$ assuming the recorded counts are correct, is given by:

$$ASN \simeq \frac{ln(1/\alpha) + 0.5ln(2s_{wl})}{(p_w ln(2s_{wl}) + p_l ln(2 - 2s_{wl}))} [8], Eqn\ (5). \qquad (3)$$

Example 2. Consider the first round of the IRV election of Example 1. If we view this first round as a FPTP election with winners c_1, c_2, and c_4, and loser c_3, the null hypotheses we need to reject are $\tilde{f}(c_1) \le \tilde{f}(c_3)$, $\tilde{f}(c_2) \le \tilde{f}(c_3)$, $\tilde{f}(c_4) \le \tilde{f}(c_3)$. We calculate $p_1 = 26000/60000$, $p_2 = 10000/60000$, $p_3 = 9000/60000$, $p_4 = 15000/60000$ and $s_{13} = 26000/35000$, $s_{23} = 10000/19000$, and $s_{43} = 15000/24000$. The ASN for rejecting each hypothesis using BRAVO, assuming $\alpha = 0.05$, is 44.5, 6885, and 246 respectively. □

4.2 Ballot-level Comparison Risk-Limiting Audits for FPTP

Stark [19] presents a method for conducting a ballot-level comparison RLA of a collection of FPTP contests or races simultaneously. In this section, we describe this audit in the context of a single race, where each ballot records a vote for a single candidate. This audit randomly samples ballots from the set $\tilde{\mathcal{B}}$ and finds the matching electronic records for those ballots in the set $\mathcal{B}$. For each ballot, we compare the actual ($\tilde{b}$) and recorded (b) representations. We assess any differences in these representations in terms of the extent to which the error overstated a pairwise margin between a winning and losing candidate. The procedure followed in this comparison RLA is shown in Figure 3.

We denote the algorithm shown in Figure 3 as MACRO. For each sampled ballot, we compute its maximum across-contest relative overstatement (MACRO [17, 18]) in the single-contest setting. In an election with winners $\mathcal{W}$, and losers $\mathcal{L}$, the MACRO for a ballot p is given by:

$$e_p = \max_{w \in \mathcal{W}, l \in \mathcal{L}} (v_{pw} - a_{pw} - v_{pl} + a_{pl})/V_{wl} \tag{4}$$

where: $v_{pc} \in \{0, 1\}$ is 1 if p is a recorded vote for candidate c, and 0 otherwise; $a_{pc} \in \{0, 1\}$ is 1 if $\tilde{p}$ is an actual vote for candidate c; and V_{ij} the pairwise margin (difference in recorded tallies) between candidates i and j.

As each ballot p is sampled, we multiply a running Kaplan-Markov MACRO P-value (P_{KM}) as follows [19]:

$$P_{KM} = P_{KM} \times \frac{1 - 1/U}{1 - \frac{e_p}{2\gamma/V_{min}}} \tag{5}$$

where: V_{min} is the smallest recorded margin between a winning and losing candidate; $U = (2\gamma|\mathcal{B}|)/V_{min}$; γ is a parameter used to inflate the upper bound on errors for each ballot (see Stark [19] for a description of the role and importance of the 'inflator' γ); and e_p is defined as per Equation 4. We continue to sample ballots until either a maximum number M of ballots have been checked (indicating that a full recount is required), or our P_{KM} statistic falls below our risk limit α.

Given an overall risk limit α, we can estimate the number of ballots that must be sampled by such an audit under the assumption that no errors are present in the electronic ballot records. We reuse the terminology of ballot polling audits, and call this number of ballots the *Average Sample Number (ASN)* for the audit.

macro($\mathcal{B}$, $\tilde{\mathcal{B}}$,$\mathcal{W}$,$\mathcal{L}$,M,α,γ)
    V_{min} := $\min_{w \in \mathcal{W}, l \in \mathcal{L}} f(w) - f(l)$
    $\mu := \frac{V_{min}}{|\mathcal{B}|}$
    $U := 2\gamma/\mu$
    $P_{KM} := 1$
    $n := 1$
    repeat
        Randomly draw a ballot $\tilde{b} \in \tilde{\mathcal{B}}$ with corresponding reported ballot $b \in \mathcal{B}$
        For each candidate c, define $v_{bc} = 1$ if $first(b) = c$ and $a_{bc} = 1$ if
            $first(\tilde{b}) = c$, and $v_{bc} = a_{bc} = 0$ otherwise
        $e_b := \max_{w \in \mathcal{W}, l \in \mathcal{L}} \frac{v_{bw} - a_{bw} - v_{bl} + a_{bl}}{f(w) - f(l)}$
        $P_{KM} = P_{KM} \times \frac{1 - 1/U}{1 - \frac{e_b}{2\gamma/V_{min}}}$
        if$(P_{KM} \leq \alpha)$
            % reported results stand
            return *true*
        $n := n + 1$
    until$(n \geq M)$
    % full recount required
    return *false*

Figure 3: A comparison RLA of a FPTP election with actual ballots $\tilde{\mathcal{B}}$, reported ballots $\mathcal{B}$, declared winners $\mathcal{W}$ and losers $\mathcal{L}$, inflation factor γ, risk limit α and sampling limit M.

Given an election with reported ballots $\mathcal{B}$, the ASN for a comparison RLA of the form shown in Figure 3, with risk limit α, is defined by Stark [19]:

$$ASN \simeq -ln(\alpha)\, U \tag{6}$$

where U is defined as above.[2]

Example 3. Consider again the first round of the IRV election of Example 1, viewed as a FPTP election with winners c_1, c_2, and c_4, and loser c_3. This election can be audited by a single application of MACRO (Figure 3). The tallies for each candidate are shown in Table 1, column two. The margins between each winner-loser pair in this first round election are $V_{c_1,c_3} = 17000$, $V_{c_2,c_3} = 1000$, and $V_{c_4,c_3} = 6000$. The smallest winner-loser margin V_{min} is 1000. Using the formula stated in Equation 6, with $\alpha = 0.05$ and $\gamma = 1.1$, the expected number of ballot checks required by MACRO is 395.4, with $U = (2\gamma|\mathcal{B}|)/V_{min} = 132$.

When auditing this first-round election, the algorithm of Figure 3 randomly draws a paper ballot $\tilde{b} \in \tilde{B}$ and compares it to its electronic record $b \in \mathcal{B}$. If $\tilde{b}$ and b match, the computed error e_b is equal to 0. Consider the situation in which a paper ballot with ranking $\tilde{b} = [c_3,\, c_4]$ has been recorded as a $b =$

[2]Equation 6 is derived from Eqn 17 on page 8 of [19] with $k = 0$.

[c_2, c_3, c_4] ballot, with the election profile listed in Example 1 representing reported counts. To determine the impact of this erroneous recorded ballot, we look at each winner w and loser l pair (c_1,c_3), (c_2, c_3), and (c_4, c_3). For each winner-loser pair, we compute, and take the maximum of, the expression:

$$\frac{v_{bw} - a_{bw} - v_{bl} + a_{bl}}{f(w) - f(l)}$$

For (c_1, c_3) and (c_4, c_3) the numerator in the above expression is equal to 1 – the error in the reported ballot overestimated the margin between these winners and the loser by 1 vote. For pair (c_2, c_3), the numerator in the above expression is equal to 2 – the error in the reported ballot overestimated the margin between c_2 and c_3 by 2 votes. For this ballot, $e_b = 2 \times 10^{-3}$. □

5 Risk-Limiting Audits for IRV Elections

In this section we present four different approaches for conducting a ballot-polling or ballot-level comparison RLA for an IRV election. The first method audits the entire elimination order, ensuring that every step in the IRV election was correct (with some confidence). The second method simplifies the auditing task in settings where we can eliminate multiple candidates in a single round. The third method seeks to examine only whether the eventual winner was the correct one. The fourth approach is a general algorithm for finding efficient ballot-polling and ballot-level comparison RLAs for IRV elections.

Each of these involves auditing simultaneously a collection $\mathcal{F}$ of different facts, whose conjunction is what we actually want to check. In the first case, we are interested in checking the complete elimination order; in later methods we are interested in a collection of facts which, taken together, imply that the announced winner truly won. Each individual audit is conducted to the same Risk Limit α. If at any point, any of the audits fail to reach a positive conclusion, we manually recount the whole election. It is easy to see that this process constitutes a valid risk-limiting audit to risk limit α of the election result, assuming that our collection of chosen facts does indeed imply that that candidate won. Suppose that the announced election outcome is actually wrong. Then at least one fact in $\mathcal{F}$ must be false. The individual audit of that fact will therefore go to a full manual recount with probability at least $1 - \alpha$, at which point we hand count the whole election.

Although the risk limit is preserved, the likelihood of unnecessarily manually recounting an election that is actually correct is higher when the conjunction of many facts is being checked.[3]

5.1 Auditing a particular elimination order

The simplest approach to applying risk limiting auditing to IRV is to consider the IRV election as a number of simultaneous FPTP elections, one for each IRV

[3]Thanks to Damjan Vukcevic for pointing this out.

```
bp-irvbravo(B̃,π,α,M)
    H := ∅
    for(i ∈ 1..|π| − 1)
        l := π(i)
        C_l := {π(i), π(i + 1), . . . , π(|π|)}
        for(j ∈ i + 1..|π|)
            w := π(j)
            T_wl := 1
            s_wl := t_{C_l}(w)/(t_{C_l}(w) + t_{C_l}(l))
            H := H ∪ {(w, l)}
    m := 0
    while(m < M ∧ H ≠ ∅)
        randomly draw ballot b from B̃
        m := m + 1
        for((w, l) ∈ H)
            if(w = first(p_{C_l}(b)))
                T_wl := T_wl × 2s_wl
                if(T_wl ≥ 1/α)
                    % reject the null hypothesis
                    H = H − {(w, l)}
            else if(l = first(p_{C_l}(b)))
                T_wl := T_wl × 2(1 − s_wl)
    if(H = ∅)
        % reported results stand
        return true
    else % full recount required
        return false
```

Figure 4: A risk-limiting ballot-polling RLA of an IRV election with actual ballots $\tilde{\mathcal{B}}$, order of elimination π, risk limit α and limit on ballots checked M.

round. This was suggested by Sarwate *et al* [14], although they do not explore it algorithmically. Note that this may perform much more auditing than required, since it verifies more than just that the eventual winner is the correct winner, but that every step in the IRV election was correct (with some confidence).

Given an election $\mathcal{B}$ of n candidates $\mathcal{C}$, we define the computed elimination order as $\pi = [c_1, c_2, \ldots, c_{n-1}, c_n]$ where c_1 is the first eliminated candidate, c_2 the second, etc, and c_n the eventual winner.

We treat each IRV round as a FPTP election. In round i, we have a set of winning candidates (C_w, the candidates that are still standing after round i) and a single losing candidate ($l = c_i$, the candidate eliminated in round i). More formally, the set of candidates in the round i FPTP election is $C_l = \{c_j \mid i \leq j \leq n\}$. Each candidate $c \in C_l$ has a recorded tally of $t_{C_l}(c)$. The loser of this election is $l = c_i$ and the set of $n - i$ winners denoted by $C_l \setminus \{l\}$.

5.1.1 Auditing the elimination order by ballot-polling

We can audit all these FPTP elections simultaneously by considering all the null hypotheses that would violate the announced result. These are $\{\tilde{t}_{C_l}(c) \leq \tilde{t}_{C_l}(c_l) \mid 1 \leq i \leq n-1, l = c_i, c \in C_i \setminus \{l\}\}$. We represent these hypotheses by a pair (w, l) of winner $w = c$, and loser $l = c_i$. The statistic maintained for this test is T_{wl}. Each loser loses in only one round so there is no ambiguity.

The algorithm is shown in Figure 4. The set of hypotheses H are again pairs (w, l) of winner w and loser l, but are interpreted as a hypothesis for the FPTP election corresponding to the round where l was eliminated. The calculation of the expected ratio of votes s_{wl} must be made using the tallies from this round, and we must consider every ballot to see how it is interesting for that particular hypothesis. A ballot that is exhausted after k rounds, for example, will not play a role when determining the statistics for later round hypotheses.

Example 4. Consider the IRV election shown in Example 1. The null hypotheses we need to reject are $\tilde{f}(c_1) \leq \tilde{f}(c_3)$, $\tilde{f}(c_2) \leq \tilde{f}(c_3)$, and $\tilde{f}(c_4) \leq \tilde{f}(c_3)$ from the first round election, $\tilde{t}_{\{c_1,c_2,c_4\}}(c_1) \leq \tilde{t}_{\{c_1,c_2,c_4\}}(c_2)$ and $\tilde{t}_{\{c_1,c_2,c_4\}}(c_3) \leq \tilde{t}_{\{c_1,c_2,c_4\}}(c_2)$ from the second round election and $\tilde{t}_{\{c_1,c_4\}}(c_4) \leq \tilde{t}_{\{c_1,c_4\}}(c_1)$ from the final round. Assuming $\alpha = 0.05$ the ASNs for the first round are the same as calculated in Example 2. The ASNs required to disprove the two stated null hypotheses for the second round election are 51.8 and 64.0. The ASN required to disprove the final round null hypothesis is 1186. The ASN of the overall audit is the maximum of the ASNs required to disprove all null hypotheses, across each round. For this election, this ASN is 6885. □

The weakness of this naive approach is that inconsequential earlier elimination rounds can be difficult to audit even if they are irrelevant to the winner.

Example 5. Consider an election with candidates c_1, c_2, c_3, c_4, c_5 and ballots $[c_1]$: 10000, $[c_2]$: 6000, $[c_3, c_2]$: 3000, $[c_3, c_1]$: 2000, $[c_4]$: 500, $[c_5]$: 499. The elimination order is $[c_5, c_4, c_3, c_2, c_1]$. Given $\alpha = 0.05$, rejecting the null hypothesis that c_5 beat c_4 in the first round gives an ASN of $13,165,239$ indicating a full hand audit is required. But it is irrelevant to the election result. □

5.1.2 Auditing the elimination order by a comparison audit

Each of these FPTP elections can also be audited via a single application of MACRO (Figure 3) with $\mathcal{W} = C_l \setminus \{l\}$, $\mathcal{L} = \{l\}$, and appropriate instantiations of the risk limit α and inflator γ parameters. As in the ballot-polling context, we can audit each of these FPTP elections simultaneously. In contrast to the ballot-polling audit, we need only perform a single comparison RLA (using MACRO) for each IRV round. Our ballot-polling audit, in contrast, applies BRAVO to each of a series of hypotheses in each round (one for each winner-loser pair).

Example 6. Consider again the IRV election shown in Example 1. To audit the entire elimination order with a comparison audit, we treat each IRV round as an FPTP election and run a MACRO audit. Assuming $\alpha = 0.05$, and $\gamma = 1.1$, the expected number of ballot checks required by MACRO is the same as that calculated in Example 3. For the remaining two IRV rounds, the ASNs required by MACRO are 28.2 and 98.9. The ASN of the overall audit is the maximum of the ASNs required by MACRO in each round. For this election, this ASN is 395.4. In this case, auditing the entire elimination order by a comparison audit is likely to be more efficient than a corresponding ballot-polling audit. □

5.2 Simultaneous elimination

It is common in IRV elections to eliminate multiple candidates in a single round if it can be shown that the order of elimination cannot affect later rounds. Given an elimination order π we can simultaneously eliminate candidates $E = \{\pi(i)..\pi(i+k)\}$ if the sum of tallies of these candidates is less than the tally of the next lowest candidate. Let $C = \{\pi(i), \pi(i+1), \ldots \pi(k), \pi(k+1), \ldots \pi(n)\}$ be the set of candidates standing after the first $i-1$ have been eliminated. We can simultaneously eliminate E if:

$$t_C(c) > \sum_{c' \in E} t_C(c') \quad \forall c \in C \setminus E \tag{7}$$

This is because no matter which order the candidates in E are eliminated no candidate could ever garner a tally greater than one of the candidates in $C \setminus E$. Hence they will all be eliminated in any case. As the remainder of the election only depends on the set of eliminated candidates and not their order, the simultaneous elimination can have no effect on later rounds of the election.

We can model the simultaneous elimination for auditing by considering all the simultaneously eliminated candidates E as as single loser l. Like the audit of a particular elimination sequence, we are proving a stronger result than necessary, *i.e.* that a particular sequence of (possibly multiple) eliminations is valid, though there may be another way of getting the same candidate to win even if the multiple elimination isn't correct.

This often results in a much lower ASN, though not necessarily: sometimes the combined total of first preferences in E is very close to the next tally, so a lot of auditing is required. It may be better to audit each elimination individually in this case. It is possible to compute the ASN for each approach and choose the method that requires the least auditing, assuming the outcome is correct.

5.2.1 Simultaneous elimination by ballot-polling

We want to reject hypotheses $\tilde{t}_C(c) \leq \tilde{t}_C(l)$ for each $c \in C \setminus E$. The statistic T_{wl} in this case is increased when we draw a ballot where w is the highest-ranked of remaining candidates C, and decreased when we draw a ballot where $c' \in E$ is the highest-ranked of remaining candidates C.

The elimination of all these null hypotheses is sufficient to prove that the multiple elimination is correct. This can then be combined with the audit of the rest of the elimination sequence, as described in Section 5.1, to test whether the election's announced winner is correct.

Example 7. Consider the election in Example 5. We can multiply eliminate the candidates $E = \{c_5, c_4\}$ since the sum of their tallies $499 + 500 < 5000$ which is the lowest tally of the other candidates (c_1, c_2, and c_3). If we do this the difficult first round elimination auditing disappears. This shows the benefit of multiple elimination. The ASNs required for the joint elimination of E are 17.0, 36.2 and 49.1 as opposed to requiring a full hand audit.

After this simultaneous elimination, the tallies for the three candidate election $\{c_1, c_2, c_3\}$ are $c_1 : 10000$, $c_2 : 6000$ and $c_3 : 5000$ and the ASNs to reject the hypotheses $\tilde{t}_{\mathcal{C}}(c_1) \leq \tilde{t}_{\mathcal{C}}(c_3)$ and $\tilde{t}_{\mathcal{C}}(c_2) \leq \tilde{t}_{\mathcal{C}}(c_3)$ are 77.6 and 1402 respectively.

We could also simultaneously eliminate $E = \{c_5, c_4, c_3\}$ since the sum of their tallies $499 + 500 + 5000 < 6000$ which is the lowest tally of the other candidate (that of c_2). But this will lead to a very difficult hypothesis to reject, $\tilde{t}_{\mathcal{C}}(c_2) \leq \tilde{t}_{\mathcal{C}}(\{c_5, c_4, c_3\})$ since the tallies are almost identical! The ASN is 158,156,493! This illustrates that simultaneous elimination may not always be beneficial. □

5.2.2 Simultaneous elimination by a ballot-level comparison RLA

As in the ballot-polling context, we treat any simultaneously eliminated candidates E as a single loser l, eliminated in a single round i. We treat each round as a FPTP election, audited via a single application of MACRO.

Example 8. Consider again the election in Example 5, in the setting where we simultaneously eliminate candidates $E = \{c_5, c_4\}$ in the first round. When viewed as a single losing candidate l, the winner-loser pairwise margins in this first round FPTP election are $V_{c_1,l} = 9001$, $V_{c_2,l} = 5001$, and $V_{c_3,l} = 4001$. Assuming $\alpha = 0.05$ and $\gamma = 1.1$, the expected number of ballot checks required by MACRO to audit this first round FPTP election is 36.2. In the second round FPTP election, candidates c_1, c_2 and c_3 remain with winners $\{c_1, c_2\}$ and loser c_3. The winner-loser pairwise margins in this election are $V_{c_1,c_3} = 5000$ and $V_{c_2,c_3} = 1000$, with $V_{min} = 1000$. The expected number of ballot checks required by MACRO to audit this election is 145. In the final round election, our winner is c_1 and loser c_2, with $V_{c_1,c_2} = 3000$. The expected number of ballot checks required by MACRO to audit this election is 48.3. The overall ASN for the comparison audit, given simultaneous elimination of candidates c_4 and c_5, is 145. This is less than that of the ballot-polling variant at 1402. □

5.3 Winner only auditing

The above two methods consider auditing the entire IRV process to ensure that we are confident on all its outcomes – i.e., that the correct candidate was eliminated in each round. This is too strong since even if earlier eliminations happened in a different order it may not have any effect on the eventual winner.

Example 9. Consider an election with ballots $[c_1, c_2, c_3] : 10000$, $[c_2, c_1, c_3] : 6000$ and $[c_3, c_1, c_2] : 5999$. No simultaneous elimination is possible, and auditing that c_3 is eliminated before c_2 will certainly require a full hand audit. But even if c_2 were eliminated first it would not change the winner of the election. □

5.3.1 Winner only auditing via ballot-polling

An alternate approach to ballot-polling RLAs for IRV elections is to simply reject the $n-1$ null hypotheses $\{\tilde{f}(w) \leq \tilde{t}_{\{w,l\}}(l)\}$ where w is the declared winner of the IRV election, and $l \in \mathcal{C} \setminus \{w\}$. This hypothesis states that l gets more votes than w where l is given the maximal possible votes it could ever achieve before w is eliminated, and w gets only its first round votes (the minimal possible votes it could ever hold). When we reject this hypothesis we are confident that there could not be any elimination order where w is eliminated before l. If all these hypotheses are rejected then we are assured that w is the winner of the election, independent of a particular elimination order.

Example 10. Consider the election of Example 9. We must reject the hypotheses that $\{\tilde{f}(c_1) \leq \tilde{t}_{\{c_1,c_2\}}(c_2)\}$ (c_1 is eliminated before c_2) and $\{\tilde{f}(c_1) \leq \tilde{t}_{\{c_1,c_2\}}(c_3)\}$ (c_1 is eliminated before c_3). The primary vote for c_1 is 10000, while the maximum tally that c_2 can achieve before c_1 is eliminated is 6000. The maximum tally that c_3 can achieve before c_1 is eliminated is 5999. Auditing to reject these hypotheses is not difficult. The ASNs are 98.4 and 98.3 ballots.

Note that if the $[c_2, c_1, c_3]$ ballots were changed to $[c_2, c_3, c_1]$, the maximum tally that c_3 can achieve is 12000, and the hypothesis that (c_1 is eliminated before c_3) could not be rejected. In this case just changing a single vote could result in c_3 winning the election, so this election will need a full recount. □

5.3.2 Winner only auditing via a ballot-level comparison RLA

The ballot-level comparison RLA version of the winner only audit proceeds in a similar fashion to its ballot-polling counterpart. Given a election with winner w and losers $\mathcal{C} \setminus \{w\}$, the ballot-polling audit executes a BRAVO audit for each winner-loser (w, l) pair, where $l \in \mathcal{C} \setminus \{w\}$. In each of these audits, w is awarded only their first preference votes $\tilde{f}(w)$, while l is awarded all votes in which they appear before w, or where they appear and w does not $\tilde{t}_{\{w,l\}}$. This audit is designed to disprove the null hypothesis that $\tilde{f}(w) \leq \tilde{t}_{\{w,l\}}$. In the ballot-level comparison RLA version of this audit, we apply the MACRO algorithm of Figure 3, in place of BRAVO, for each winner-loser pair (w, l), with $\mathcal{W} = \{w\}$ and $\mathcal{L} = \{l\}$.

Example 11. Consider the election of Example 9. For winner-loser pair (c_1, c_3), we apply MACRO to an election with winner c_1, and loser c_3, where c_1 has a tally of 10000 votes and c_3 a tally of 5999 votes. Even though c_1 appears before c_3 in the [c_2, c_1, c_3] ballots, we only award c_1 with its first preference votes in a winner only audit. If the positions of c_1 and c_3 were swapped in these ballots, these ballots would be treated as votes for c_3. In this election, $V_{min} = 4001$ and we expect to check 36.2 ballots. For winner-loser pair (c_1, c_2), we apply MACRO to an election with winner c_1, and loser c_2, where c_1 has a tally of 10000 votes and c_2 a tally of 6000. The ASN for this election is also 36.2 ballots. □

6 A general algorithm for finding efficient ballot-polling or comparison RLAs for IRV

In each of the ballot-polling and ballot-level comparison RLAs for IRV described in the preceding sections, we apply an existing risk limiting audit (BRAVO, as per Figure 4, or MACRO, as per Figure 3) to confirm a collection of *facts* with a given level of statistical confidence. In the case of a winner only audit, for example, we are seeking to confirm that the reported winner w could not have been eliminated before any one of the reported losers $l \in \mathcal{C} \setminus \{w\}$. This results in $|\mathcal{C} \setminus \{w\}|$ facts to be confirmed, one for each winner-loser pair.

For each fact that we seek to confirm, we can estimate the number of ballots that must be checked via an application of BRAVO or MACRO, assuming no errors are found. We present a general algorithm for choosing the set of facts h that can be checked most efficiently to confirm that the reported winner c_w was the correct one. The algorithm, denoted *audit-irv*, achieves this by finding the easiest way to show that all election outcomes in which a candidate other than c_w won, did not arise, with a given level of statistical confidence, for a given method of auditing each fact. The *audit-irv* algorithm can be applied to generate either a ballot-polling or a ballot-level comparison RLA for an IRV election.

Note that our *risk-limit* follows directly from BRAVO and MACRO: if the election outcome is wrong, then one of the facts in h must be false—a BRAVO or MACRO audit with risk limit α will detect this with probability at least $1 - \alpha$, and we then manually recount the whole election. However, our estimate of *efficiency* is only heuristic: ASNs for testing a single fact can be derived analytically, but the expected number of samples required to reject multiple hypothesis at once is very hard to compute, even if there are no discrepancies. We make a best guess based on the maximum ASN for any single fact—this is what we meant by "optimal" in this section, though it may not guarantee an optimally efficient audit overall.

Our algorithm, *audit-irv*, outlined in Figure 5, explores the tree of alternate elimination sequences, ending in a candidate $c' \neq c_w$. Each node is a partial (or complete) elimination sequence. For each node π, we consider the set of hypotheses that (i) can be proven with an application of BRAVO or MACRO and (ii) any one of which disproves the outcome that π represents. We label each node π with the hypothesis h from this set that requires the least number of anticipated ballot samples (ASN) to prove, denoted $asn(h)$. We use the notation $h(\pi)$ and $asn(\pi)$ to represent the hypothesis assigned to π and the ASN for this hypothesis, respectively. Our algorithm finds a set of hypotheses to prove, denoted *audits*, that: validates the correctness of a given election outcome, with risk limit α; and for which the largest ASN of these hypotheses is minimised. When performing a ballot-polling audit, we compute this ASN via Equation 3, and Equation 6 when generating a comparison RLA.

Consider a partial elimination sequence $\pi = [c, \ldots, w]$ of at least two candidates, leading to an alternate winner w. This sequence represents the suffix of a complete order – an outcome in which the candidates in $\mathcal{C} \setminus \pi$ have been previously eliminated, in some order. We define a function $\mathsf{FindBestAudit}(\pi, \mathcal{C}, \mathcal{B}, \alpha[, \gamma])$ that finds the easiest to prove hypothesis (or fact) h, with the smallest ASN, which disproves the outcome π given risk limit α. The parameter γ is only used when generating a ballot-level comparison RLA to audit the given election. For the outcome $\pi = [c|\ldots]$, $\mathsf{FindBestAudit}$ considers the following hypotheses:

WO(c,c'): Hypothesis that c beats $c' \in \pi$, for some $c' \in \pi, c' \neq c$, in a winner only audit of the form described in Section 5.3, with winner c and loser c', thus invalidating the sequence since c cannot be eliminated before c';

WO(c'',c): Hypothesis that $c'' \in \mathcal{C} \setminus \pi$ beats c in a winner only audit with winner c'' and loser c, thus invalidating the sequence since c'' cannot be eliminated before c;

IRV(c,c',$\{c'' \mid c'' \in \pi\}$): Hypothesis that c beats some $c' \neq c \in \pi$ in a BRAVO (or MACRO) audit with winner c and loser c', under the assumption that the only candidates remaining are those in π (i.e. the set $\{c'' \mid c'' \in \pi\}$) with other candidates eliminated with their votes distributed to later preferences, thus invalidating the sequence since then c is not eliminated at this stage in an IRV election.

We assume that if no hypothesis exists with ASN less than $|\mathcal{B}|$ the function returns a dummy **INF** hypothesis with $ASN(\mathbf{INF}) = +\infty$.

For an election with candidates $\mathcal{C}$ and winner c_w, *audit-irv* starts by adding $|\mathcal{C}| - 1$ partial elimination orders to an initially empty priority queue F, one for each alternate winner $c \neq c_w$ (Steps 4 to 9). The set *audits* is initially empty. For orders π containing a single candidate c, $\mathsf{FindBestAudit}$ considers the hypotheses $\mathbf{WO}(c'',c)$, candidate $c'' \neq c$ beats c in a winner only audit of the form described in Section 5.3, with winner c'' and loser c, for each $c'' \in \mathcal{C} \setminus \{c\}$. The hypothesis h with the smallest $ASN(h)$ is recorded in $hy[\pi]$. The (current) best ancestor for π is recorded in $ba[\pi]$, for these singletons sequences it is always the sequence itself.

We repeatedly find and remove a partial sequence π in F for expansion (Steps 11 and 12). This is the sequence with the (equal) highest ASN. If the best ancestor for this sequence has an ASN lower than the current lower bound LB (Steps 13 to 16) we simply add the corresponding hypothesis to *audits* and remove any sequences in F which are subsumed by this ancestor (have it as a suffix), and restart the main loop.

Otherwise (Steps 17 to 31) we create a new elimination sequence π' with c appended to the start of π ($[c] +\!\!+ \pi$) for each $c \in \mathcal{C} \setminus \pi$. For a new sequence π', $\mathsf{FindBestAudit}$ finds the hypothesis h requiring the least auditing effort to prove. We record (Step 20) this as the hypothesis for $hy[\pi'] = h$.

We calculate (Step 21) the best ancestor of π' by comparing the ASN for its hypothesis with that of its ancestor. If the sequence π' is complete, then we known one of its ancestors (including itself) must be audited. If the best of these is infinite, we terminate, a full recount is necessary. Otherwise we add the

```
audit-irv(\mathcal{C}, \mathcal{B}, c_w, \alpha[, \gamma])
audits \leftarrow \emptyset
F \leftarrow \emptyset ▷ F is a set sequences to expand (the frontier)
LB \leftarrow 0
▷ Populate F with single-candidate sequences
for each(c \in \mathcal{C} \setminus \{c_w\}):
    \pi \leftarrow [c]
    h \leftarrow FindBestAudit(\pi, \mathcal{C}, \mathcal{B}, \alpha[, \gamma])
    hy[\pi] \leftarrow h ▷ Record best hypothesis for \pi
    ba[\pi] \leftarrow \pi ▷ Record best ancestor sequence for \pi
    F \leftarrow F \cup \{\pi\}
▷ Repeatedly expand the sequence with largest ASN in F
while(|F| > 0):
    \pi \leftarrow argmax\{ASN(hy[\pi]) | \pi \in F\}
    F \leftarrow F \setminus \{\pi\}
    if(ASN(hy[ba[\pi]]) \leq LB):
        audits \leftarrow audits \cup \{hy[ba[\pi]]\}
        F \leftarrow F \setminus \{\pi' \in F | ba[\pi] is a suffix of \pi'\}
        continue
    for each(c \in \mathcal{C} \setminus \pi):
        \pi' \leftarrow [c] ++ \pi
        h \leftarrow FindBestAudit(\pi', \mathcal{C}, \mathcal{B}, \alpha[, \gamma])
        hy[\pi'] \leftarrow h
        ba[\pi'] \leftarrow if ASN(h) < ASN(hy[ba[\pi]]) then \pi' else ba[\pi]
        if(|\pi'| = |\mathcal{C}|):
            if(ASN(hy[ba[\pi']]) = \infty):
                terminate algorithm, full recount necessary
            else:
                audits \leftarrow audits \cup \{hy[ba[\pi']]\}
                LB \leftarrow max(LB, ASN(hy[ba[\pi']]))
                F \leftarrow F \setminus \{\pi' \in F | ba[\pi] is a suffix of \pi'\}
                continue
        else:
            F \leftarrow F \cup \{\pi'\}
return audits with maximum ASN equal to LB
```

Figure 5: The *audit-irv* algorithm for searching for a collection of hypothesis to audit, with parallel applications of BRAVO or MACRO, that define a risk-limiting audit of an IRV election with candidates $\mathcal{C}$, ballots $\mathcal{B}$, and winner c_w, with a given risk limit α, and inflator parameter γ (relevant if constructing a comparison RLA).

hypothesis of its best ancestor to *audits* and remove all sequences in F which are subsumed by this ancestor. If the sequence is not complete we simply add it into the set of sequences to be expanded F.

Example 12. Consider an election with ballots $[c_1, c_2, c_3]$: 5000, $[c_1, c_3, c_2]$: 5000 $[c_2, c_3, c_1]$: 5000, $[c_2, c_1, c_3]$: 1500, $[c_3, c_2, c_1]$: 5000, $[c_3, c_1, c_2]$: 500, and $[c_4, c_1]$: 5000, and candidates c_1 to c_4. The initial tallies are: c_1: 10000; c_2: 6500; c_3: 5500; c_4: 5000. Candidates c_4, c_3, and c_2 are eliminated, in that order, with winner c_1. In a ballot-polling or comparison winner only audit ($\alpha = 0.05$), we cannot show that c_1 beats c_3, or that c_1 beats c_2, as c_1's first preference tally (of 10000 votes) is less than the total number of ballots that we could attribute to c_2 and c_3 (11500 and 10500, respectively). Simultaneous elimination is not applicable in this instance, as no sequences of candidates can be eliminated in a group. In an audit of the whole elimination order (as per Section 5.1), the loss of c_4 to c_1, c_2, and c_3 is the most challenging to audit. The ASN for the ballot-polling version of this audit, assuming $\alpha = 0.05$, is 25% of all ballots (6750 ballots). The comparison version of this audit, assuming $\gamma = 1.1$, is 1.3% of all ballots (351 ballots).

Our *audit-irv* algorithm finds a set of hypotheses that can be proven using a ballot-polling audit with a maximum ASN of 1% (or 270 ballots, with $\alpha = 0.05$), and that consequently rule out all elimination sequences that end in a candidate other than c_1. This audit tests the hypotheses: c_1 beats c_2 if c_3 and c_4 have been eliminated (ASN of 1%); c_1 beats c_3 if c_2 and c_4 have been eliminated (ASN 0.5%); c_1 beats c_4 in a winner only audit (ASN 0.4%); and that c_1 beats c_3 if c_4 has been eliminated (ASN 0.1%). If we instead use *audit-irv* to construct a ballot-level comparison audit, we find a set of hypotheses that can be tested with a maximum ASN of 0.17%. These hypotheses are: c_1 beats c_2 if c_3 and c_4 have been eliminated (ASN of 0.17%); c_1 beats c_3 if c_4 has been eliminated (ASN of 0.07%) and if c_2 and c_4 have been eliminated (ASN of 0.11%); c_1 beats c_4 in a winner only audit (ASN of 0.13%); and that both c_2 and c_3 beat c_4 if all remaining candidates have been eliminated (ASN of 0.04%). □

7 Computational Results

We have simulated the ballot-polling and ballot-level comparison RLAs described in Section 5.1 (auditing the elimination order, EO), Section 5.2 (auditing with simultaneous elimination, SE), and Section 5.3 (winner only auditing, WO), on 21 US IRV elections held between 2007 and 2014, and on the IRV elections held across 93 electorates in the 2015 state election in New South Wales (NSW), Australia. For each election, we have simulated each of these audits with varying risk limits ($\alpha = 1\%$ and $\alpha = 5\%$), and $\gamma = 1.1$.[4] We record, for each simulated audit, the number of ballots that were sampled during the audit (expressed as a percentage of ballots cast). An audit that needs to sample fewer ballots before confirming the correctness of the reported outcome, to the given degree of statistical confidence, is a more efficient audit. As each audit involves

[4]We explore the influence of the γ parameter in subsequent experiments.

ballots being drawn at random, we simulate each audit 10 times and compute the average number of ballots checked across those 10 simulations.

All experiments have been conducted on a machine with an Intel Xeon Platinum 8176 chip (2.1GHz), and 1TB of RAM.

Table 2 compares the number of ballot checks required by ballot-polling and ballot-level comparison audits of the form described in Section 5.1 across our suite of election instances. The number of required ballot samples is reported alongside the ASN for each audit (computed as per Equation 3 for each ballot-polling audit, and Equation 6 for each comparison audit), and the margin of victory (MOV) for each election (computed using the algorithm of Blom *et al* [4]). Tables 3 and 4 similarly compare the number of ballot samples required by simultaneous elimination and winner-only ballot-polling and comparison audits, described in Sections 5.2 and 5.3. In this experiment, no errors or discrepancies have been injected into the set of reported ballots in each election instance ($\mathcal{B} \equiv \tilde{\mathcal{B}}$).

Tables 2 to 4 show that performing a winner only audit can be much easier than auditing the full elimination order (with or without the use of simultaneous elimination), irrespective of whether we are conducting a ballot-polling or comparison audit. This is the case for the 2013 Minneapolis Mayor and 2014 Oakland Mayor elections. In some cases, winner only audits are more challenging (or not possible) as we seek to show that a candidate c (on just their first preference votes) could have beaten another c' (who is given all votes in which they appear before c or in which they appear, but c does not). Even if c does beat c' in the true outcome of the election, this audit may not be able to prove this (see Pierce 2008 County Executive, Oakland 2012 D5 City Council, and Aspen 2009 Mayor for examples).

Auditing with simultaneous elimination (grouping several eliminated candidates into a single 'super' candidate) can be more efficient than auditing each individual elimination. This is evident in the context of both ballot-polling audits (see Berkeley 2010 D8 City Council, Berkeley 2012 Mayor, Oakland 2010 Mayor, San Francisco 2007 Mayor, and Sydney NSW) and ballot-level comparison audits (see Balmain NSW 2015, Sydney NSW 2015, Oakland 2010 Mayor, San Leandro 2010 Mayor, and Berkeley 2010 D8 City Council). Across the 26 election instances in Tables 2 and 3, conducting a comparison audit with simultaneous elimination was beneficial in 15 instances and detrimental in 2. In the context of ballot-polling audits, simultaneous elimination was beneficial in 8 and detrimental in 5. In some instances, the tally of the super candidate is quite close to that of the next eliminated candidate, resulting in a more challenging audit. This is particularly evident when simulating a ballot-polling audit of the Campbelltown NSW and Berkeley 2010 D4 City Council elections.

Tables 2 to 4 show that comparison audits are generally more efficient than their ballot-polling counterparts, as they are for the underlying FPTP election. The Oakland 2012 D3 City Council election is an excellent example. Neither auditing the entire elimination sequence, the sequence with simultaneous elimination, or conducting a winner-only audit, is successful in the ballot-polling context. The ASN is more than the total number of ballots in each case. We can

Auditing the Entire Elimination Order via a Ballot Polling (BP) and Comparison (CP) Audit

					$\alpha = 1\%$				$\alpha = 5\%$			
					BP		CP ($\gamma = 1.1$)		BP		CP ($\gamma = 1.1$)	
	Election	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{B}\rvert$	**MOV**	Polls %	ASN %	Polls %	ASN %	Polls %	ASN %	Polls %	ASN %
1	Berkeley 2010 D7 CC	4	4,682	364 (7%)	6.7	7.2	**1.6**	**1.7**	3.9	4.7	**1.1**	**1.1**
2	Berkeley 2010 D8 CC	4	5,333	878 (16%)	∞	∞	∞	∞	∞	∞	**94.1**	**94.2**
3	Oakland 2010 D6 CC	4	14,040	2,603 (19%)	4.0	4.4	**1**	**1**	3	2.9	**0.7**	**0.7**
4	Pierce 2008 CC	4	43,661	2,007 (5%)	3.1	2.2	**0.2**	**0.3**	1.8	1.4	**0.2**	**0.2**
5	Pierce 2008 CAD	4	159,987	8,396 (5%)	0.3	0.5	**0.1**	**0.1**	0.2	0.3	**0.1**	**0.1**
6	Aspen 2009 Mayor	5	2,544	89 (4%)	62.4	71.8	**9**	**9**	52.7	46.9	**5.9**	**5.9**
7	Berkeley 2010 D1 CC	5	6,426	1,174 (18%)	**2.4**	**1.7**	5.8	5.9	**1.6**	**1.1**	3.8	3.8
8	Berkeley 2010 D4 CC	5	5,708	517 (9%)	7.5	7	**6.3**	**6.3**	6	4.7	**4.1**	**4.1**
9	Oakland 2012 D5 CC	5	13,482	486 (4%)	11.2	10.3	**1.5**	**1.5**	7.3	6.7	**1**	**1**
10	Pierce 2008 CE	5	312,771	2,027 (1%)	11.6	15.1	**0.2**	**0.2**	7.6	9.8	**0.2**	**0.2**
11	San Leandro 2012 D4 CC	5	28,703	2,332 (8%)	9.3	9.7	**0.9**	**1**	6.3	6.3	**0.6**	**0.6**
12	Oakland 2012 D3 CC	7	26,761	386 (1%)	∞	∞	**22.5**	**22.5**	∞	∞	**14.6**	**14.7**
13	Pierce 2008 CAS	7	312,771	1,111 (0.4%)	∞	∞	**3.7**	**3.7**	∞	∞	**2.4**	**2.4**
14	San Leandro 2010 Mayor	7	23,494	116 (0.5%)	∞	∞	**18.4**	**18.4**	92.9	∞	**12**	**12**
15	Berkeley 2012 Mayor	8	57,492	8,522 (15%)	94.6	∞	**5.9**	**5.9**	77	∞	**3.8**	**3.8**
16	Oakland 2010 D4 CC	8	23,884	2,329 (10%)	∞	∞	**7.7**	**7.7**	76.4	∞	**5.4**	**5.4**
17	Aspen 2009 CC	11	2,544	35 (1%)	∞	∞	∞	∞	∞	∞	∞	∞
18	Oakland 2010 Mayor	11	122,268	1,013 (1%)	∞	∞	**12.2**	**12.2**	∞	∞	**7.9**	**7.9**
19	Oakland 2014 Mayor	11	101,431	10,201 (10%)	∞	∞	∞	∞	∞	∞	∞	∞
20	San Francisco 2007 Mayor	18	149,465	50,837 (34%)	∞	∞	∞	∞	∞	∞	∞	∞
21	Minneapolis 2013 Mayor	36	79,415	6,949 (9%)	∞	∞	∞	∞	∞	∞	∞	∞
22	Balmain NSW 2015	7	46,952	1,731 (3.7%)	∞	∞	**34.9**	**34.9**	∞	∞	**22.7**	**22.7**
23	Campbelltown NSW 2015	5	45,124	3,096 (6.9%)	13.6	12.2	**2.2**	**2.2**	8.4	8	**1.4**	**1.4**
24	Gosford NSW 2015	6	48,259	102 (0.2%)	∞	∞	**6.6**	**6.6**	∞	∞	**4.3**	**4.3**
25	Lake Macquarie NSW 2015	7	47,698	4,253 (8.9%)	27.7	22.8	**3.5**	**3.5**	14.5	15	**2.3**	**2.3**
26	Sydney NSW 2015	8	42,747	2,864 (6.7%)	∞	∞	**59.6**	**59.6**	∞	∞	**38.8**	**38.8**

Table 2: Average # of ballots sampled (as a percentage of ballots cast) over 10 simulated ballot-polling (BP) and ballot-level comparison audits (CP) of 26 IRV elections using the EO (auditing the entire elimination order) method. Parameter α ranges between 1% and 5%, and $\gamma = 1.1$. Also reported is each election's margin of victory (MOV). The notation ∞ indicates a percentage of ballots (or ASN) greater than 100%. CC, CE, CAD, and CAS denote City Council, County Executive, County Auditor, and County Assessor. The most efficient audit (for α =1% and 5%) is highlighted in bold.

Auditing with Simultaneous Elimination via a BP/CP Audit

#	$\lvert\mathcal{B}\rvert$	$\alpha = 1\%$				$\alpha = 5\%$			
		BP (%)		CP ($\gamma = 1.1$, %)		BP (%)		CP ($\gamma = 1.1$, %)	
		Polls	ASN	Polls	ASN	Polls	ASN	Polls	ASN
1	4,682	7.5	7.2	**1.3**	**1.4**	4	4.7	**0.9**	**0.9**
2	5,333	2.9	4.2	**1**	**1**	2	2.8	**0.6**	**0.7**
3	14,040	0.7	0.9	**0.3**	**0.3**	0.5	0.6	**0.2**	**0.2**
4	43,661	3.1	2.2	**0.3**	**0.3**	1.8	1.4	**0.2**	**0.2**
5	159,987	0.3	0.5	**0.1**	**0.1**	0.2	0.3	**0.04**	**0.04**
6	2,544	62.4	71.8	**5.7**	**5.7**	54.8	46.9	**3.7**	**3.7**
7	6,426	2.4	1.7	**0.5**	**0.6**	1.6	1.1	**0.4**	**0.4**
8	5,708	28.7	40.7	**3**	**3.1**	17.8	26.6	**2**	**2**
9	13,482	15.1	10.3	**1.1**	**1.1**	11.8	6.7	**0.7**	**0.7**
10	312,771	11.6	15.1	**0.2**	**0.2**	7.6	9.8	**0.2**	**0.2**
11	28,703	9.3	9.7	**0.9**	**1**	6.3	6.3	**0.6**	**0.6**
12	26,761	∞	∞	**22.5**	**22.5**	∞	∞	**14.7**	**14.7**
13	312,771	∞	∞	**3.7**	**3.7**	∞	∞	**2.4**	**2.4**
14	23,494	∞	∞	**4.4**	**4.4**	92.9	∞	**2.8**	**2.8**
15	57,492	2.3	2.6	**0.2**	**0.2**	1.6	1.7	**0.2**	**0.2**
16	23,884	∞	∞	**7.7**	**7.7**	∞	∞	**5**	**5**
17	2,544	∞	∞	∞	∞	∞	∞	**82.4**	**82.4**
18	122,268	21.5	23.8	**0.6**	**0.6**	15	15.5	**0.4**	**0.4**
19	101,431	∞	∞	∞	∞	∞	∞	∞	∞
20	149,465	0.03	0.03	**0.01**	**0.01**	0.02	0.02	**0.01**	**0.01**
21	79.415	∞	∞	∞	∞	∞	∞	∞	∞
22	46,952	83.8	∞	**2.2**	**2.2**	65.4	82	**1.4**	**1.4**
23	45,124	∞	∞	**3.7**	**3.7**	∞	∞	**2.4**	**2.4**
24	48,259	∞	∞	**5**	**5**	∞	∞	**3.2**	**3.2**
25	47,698	6.9	7.8	**0.6**	**0.6**	3.2	5.1	**0.4**	**0.4**
26	42,747	3.3	4.6	**0.4**	**0.4**	2.2	3	**0.3**	**0.3**

Table 3: Average # of ballots sampled (as a percentage of ballots cast) over 10 simulated ballot-polling (BP) and comparison audits (CP) of 26 IRV elections with simultaneous elimination (SE), $\alpha \in \{1\%, 5\%\}$, and $\gamma = 1.1$. A ∞ indicates a percentage of ballots (or ASN) greater than 100%. The name, candidates, and MOV of each election are shown in Table 2. The most efficient audit (for α =1% and 5%) is highlighted in bold.

conduct a comparison audit, using each of these methods, however, that requires only a fraction of cast ballots to be sampled (23% or 6155 ballots, 23%, and 0.1% or 268 ballots, when auditing the entire elimination order, auditing with simultaneous elimination, and conducting a winner-only audit, respectively). For each simulated audit, increasing the risk limit reduced the average number of required ballot samples, as expected.

Table 5 reports the average number of ballots examined by the ballot-polling and ballot-level comparison audits generated by our *audit-irv* algorithm across the 26 considered IRV elections (with $\alpha = 5\%$). We compare this level of auditing effort against the number of ballot checks required by the best alternate auditing method (auditing the entire elimination order [EO], simultaneous elimination [SE], and winner-only auditing [WO]). Recall that *audit-irv* finds an appropriate set of facts to audit (via ballot-polling or a comparison audit)

Winner-Only Auditing via a BP/CP Audit

#	$\lvert\mathcal{B}\rvert$	$\alpha = 1\%$				$\alpha = 5\%$			
		BP (%)		CP ($\gamma = 1.1$, %)		BP (%)		CP ($\gamma = 1.1$, %)	
		Polls	ASN	Polls	ASN	Polls	ASN	Polls	ASN
1	4,682	8.7	22.4	**2.5**	**2.6**	4.9	14.7	**1.7**	**1.7**
2	5,333	1.3	1.8	**0.7**	**0.7**	0.8	1.2	**0.4**	**0.5**
3	14,040	0.4	0.5	**0.2**	**0.2**	0.3	0.3	**0.1**	**0.1**
4	43,661	3.2	4.1	**0.3**	**0.3**	1.8	2.7	**0.2**	**0.2**
5	159,987	0.5	1.2	**0.1**	**0.1**	0.3	0.8	**0.1**	**0.1**
6	2,544	∞	∞	∞	∞	∞	∞	∞	∞
7	6,426	1.1	1.1	**0.4**	**0.5**	0.8	0.7	**0.3**	**0.3**
8	5,708	4.9	7.3	**1.3**	**1.4**	3.8	4.8	**0.9**	**0.9**
9	13,482	∞	∞	∞	∞	∞	∞	∞	∞
10	312,771	∞	∞	∞	∞	∞	∞	∞	∞
11	28,703	1.1	4.4	**0.5**	**0.5**	0.8	2.9	**0.3**	**0.3**
12	26,761	∞	∞	∞	∞	∞	∞	∞	∞
13	312,771	∞	∞	∞	∞	∞	∞	∞	∞
14	23,494	∞	∞	∞	∞	∞	∞	∞	∞
15	57,492	0.2	0.2	**0.1**	**0.1**	0.1	0.2	**0.1**	**0.1**
16	23,884	0.9	3.1	**0.5**	**0.5**	0.6	2	**0.3**	**0.3**
17	2,544	∞	∞	∞	∞	∞	∞	∞	∞
18	122,268	∞	∞	∞	∞	∞	∞	∞	∞
19	101,431	0.8	19.8	**0.6**	**0.6**	0.5	12.9	**0.4**	**0.4**
20	149,465	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
21	79,415	0.5	3.1	**0.3**	**0.3**	0.3	2.1	**0.2**	**0.2**
22	46,952	5.2	31.6	**1**	**1**	3.7	20.6	**0.7**	**0.7**
23	45,124	1.3	1.7	**0.2**	**0.2**	0.9	1.1	**0.1**	**0.1**
24	48,259	∞	∞	∞	∞	∞	∞	∞	∞
25	47,698	0.7	1.6	**0.2**	**0.2**	0.5	1	**0.1**	**0.1**
26	42,747	1.6	6.9	**0.5**	**0.5**	1	4.5	**0.3**	**0.3**

Table 4: Average # of ballots sampled (as a percentage of ballots cast) over 10 simulated ballot-polling (BP) and ballot-level comparison audits (CP) of 26 IRV elections using the winner-only (WO) method, $\alpha \in \{1\%, 5\%\}$, and $\gamma = 1.1$. A ∞ indicates a percentage of ballots (or ASN) greater than 100%. The name, candidates, and MOV of each election are shown in Table 2. The most efficient audit (for α =1% and 5%) is highlighted in bold.

that, if shown to hold with a given degree of statistical confidence, confirms the reported election outcome with that degree of statistical confidence. The *audit-irv* algorithm finds the set of such facts requiring the least anticipated number (ASN) of ballot checks to confirm. Table 5 shows that while the ASN of the *audit-irv* audits is minimal – the actual level of auditing effort required by these audits will differ from these estimates, and may be greater than that required by an EO, SE, or WO audit. For ballot polling audits the discrepancy can be large. In these experiments we have not introduced any errors or discrepancies between the electronic ballot records and the paper ballots. In this setting, the ASN computed for a ballot-level comparison audit accurately represents the actual number of ballot checks or polls made during the audit.

In all but one of the election instances in Table 5, *audit-irv* is able to compute an audit configuration in less than 1 minute. The algorithm requires between

0.003s and 106s to find an audit configuration in the ballot-polling context, and 0.002s to 139s in the comparison audit context. The most time consuming instance is the 2014 Oakland Mayoral election, with *audit-irv* requiring 106s and 139s to find the best ballot-polling and comparison audit, respectively.

In 22/26 of the election instances of Table 5, the ballot-polling audit generated by *audit-irv* required a similar number of ballot samples to that of the best alternate method (EO, SE, or WO). In the remaining 4 instances, the *audit-irv* audit was significantly more efficient. Consider instances 12 and 13 – the Oakland 2012 City Council election for District 3, and the Pierce 2008 County Assessor election. For these instances, neither the EO, SE, or WO methods were able to audit the election without manually recounting all ballots. The *audit-irv* audits, however, were able to confirm the reported outcomes in these elections by sampling no more than 17% of the cast ballots, on average, when simulated. The comparison audits generated by *audit-irv* are significantly more efficient than their ballot-polling counterparts, across our suite of election instances.

Consider instance 17 in Table 5 – the Aspen 2009 City Council election. We can, with a comparison audit generated with *audit-irv*, confirm the reported outcome (with risk limit $\alpha = 5\%$) by sampling just under 10% of the cast ballots (254 ballots), on average. If we were to use one of the EO, SE, or WO approaches of conducting a comparison audit, in place of the audit generated by *audit-irv*, we would need to sample just under 83% of the cast ballots (2112 ballots), on average. Table 5 also shows that as the γ parameter increases, the number of ballots checked in a comparison audit may increase slightly, but not significantly.

We have shown that *audit-irv* is able to find efficient ballot-polling and ballot-level comparison audit configurations across a range of election instances, in the context where electronic ballot records exactly match their corresponding paper ballot. We now consider the effectiveness of our audits in the setting where varying numbers of errors (or discrepancies) are introduced into the reported (electronic or digitised) ballot records. We introduce discrepancies between reported and actual ballots according to a defined error rate, which we vary between 1% and 10%. This means that for any given ballot, there is a 1% to 10% probability that its electronic version differs, in some way, from the paper version. Recall that the electronic record of a paper ballot is a partial or complete sequence of candidates, ordered according to voter preference. We introduce an error in a reported ballot record with one of the following operations: replacing a randomly selected candidate in this preference ordering with a randomly selected candidate that does not appear in the ordering; inserting a randomly selected candidate that does not appear in the ordering into a randomly selected position; flipping the positions of two randomly selected candidates in the ordering; removing a randomly selected candidate in the ordering. For each reported ballot, we roll a die to determine whether to introduce an error. When introducing an error, we uniformly randomly choose one of the above manipulations to perform.

In this setting, we simulate each auditing approach 50 times – with 10 different seeds used to inject errors into electronic (reported) ballot records, and 5

Auditing using audit-irv via ballot-polling (BP) and comparison (CP) audits, $\alpha = 5\%$

#	$\lvert\mathcal{B}\rvert$	BP				CP							
		Best Alt. EO/SE/WO		*audit-irv*		Best Alt. $\gamma = 1.1$ EO/SE/WO		$\gamma = 1.1$ *audit-irv*		$\gamma = 1.2$ *audit-irv*		$\gamma = 1.3$ *audit-irv*	
		Method	Polls %	Polls %	ASN %	Method	Polls %	Polls %	ASN %	Polls %	ASN %	Polls %	ASN %
1	4,682	EO	3.9	5.4	4.7	SE	0.9	0.9	0.9	1	1	1.1	1.1
2	5,333	WO	0.8	0.9	0.9	WO	0.4	0.4	0.4	0.4	0.4	0.4	0.5
3	14,040	WO	0.3	0.3	0.3	WO	0.1	0.1	0.1	0.1	0.1	0.1	0.2
4	43,661	EO,SE	1.8	1.5	1.4	SE,EO	0.2	0.2	0.2	0.2	0.2	0.2	0.2
5	159,987	EO,SE	0.2	0.3	0.3	SE	0.04	0.04	0.04	0.04	0.04	0.1	0.1
6	2,544	EO	52.7	28.1	46.9	SE	3.7	3.7	3.8	4	4.1	4.4	4.4
7	6,426	WO	0.8	0.6	0.6	WO	0.3	0.3	0.3	0.3	0.3	0.3	0.3
8	5,708	WO	3.8	1.6	2.7	WO	0.9	0.6	0.7	0.7	0.7	0.7	0.8
9	13,482	EO	7.3	5.2	6.7	SE	0.7	0.7	0.7	0.7	0.7	0.8	0.8
10	312,771	EO,SE	7.6	13.9	9.8	SE,EO	0.2	0.2	0.2	0.2	0.2	0.2	0.2
11	28,703	WO	0.8	0.8	0.6	WO	0.3	0.1	0.1	0.2	0.2	0.2	0.2
12	26,761	–	∞	14.2	13.1	SE,EO	14.6	0.9	0.9	0.9	0.9	1	1
13	312,771	–	∞	17	22.7	SE,EO	2.4	0.3	0.3	0.3	0.3	0.4	0.4
14	23,494	EO,SE	92.9	87.6	∞	SE	2.8	2.8	2.8	3.1	3.1	3.4	3.4
15	57,492	WO	0.1	0.1	0.1	WO	0.1	0.04	0.04	0.04	0.04	0.04	0.1
16	23,884	WO	0.6	0.6	0.5	WO	0.3	0.1	0.1	0.2	0.2	0.2	0.2
17	2,544	–	∞	∞	∞	SE	82.4	9.4	9.5	10.2	10.3	11.1	11.1
18	122,268	SE	15	15.3	15.5	SE	0.4	0.3	0.3	0.4	0.4	0.4	0.4
19	101,431	WO	0.5	5.4	0.1	WO	0.4	0.04	0.04	0.04	0.04	0.04	0.04
20	149,465	WO	0.01	0.01	0.01	SE,WO	0.01	0.01	0.01	0.01	0.01	0.01	0.01
21	79,415	WO	0.3	0.2	0.2	WO	0.2	0.1	0.1	0.1	0.1	0.1	0.1
22	46,952	WO	3.7	3.2	1.9	WO	0.7	0.2	0.2	0.2	0.2	0.2	0.2
23	45,124	WO	0.9	0.8	0.7	WO	0.1	0.1	0.1	0.1	0.1	0.1	0.1
24	48,259	–	∞	∞	∞	SE	3.2	3.2	3.2	3.5	3.5	3.8	3.8
25	47,698	WO	0.5	0.5	0.3	WO	0.1	0.1	0.1	0.1	0.1	0.1	0.1
26	42,747	SE	1	1.3	0.7	SE	0.3	0.1	0.1	0.1	0.1	0.1	0.1

Table 5: Comparison of the best ballot-polling and ballot-level comparison audit methods across 26 IRV elections. We compare the average number of ballot samples required (expressed as a percentage of total ballots cast) by the best alternate ballot-polling or comparison audit methods (EO, SE, and WO) and those generated by *audit-irv*. The notation ∞ indicates a percentage of ballots (or ASN) greater than 100%.

Auditing using audit-irv via ballot polling (BP) and comparison (CP) audits, $\alpha = 5\%$, 1% errors

#	$\lvert\mathcal{B}\rvert$	BP				CP							
		Best Alt. EO/SE/WO		*audit-irv*		Best Alt. $\gamma = 1.1$ EO/SE/WO		$\gamma = 1.1$ *audit-irv*		$\gamma = 1.2$ *audit-irv*		$\gamma = 1.3$ *audit-irv*	
		Method	Polls %	Polls %	ASN %	Method	Polls %	Polls %	ASN %	Polls %	ASN %	Polls %	ASN %
1	4,682	EO	3.7	3.8	4.8	SE	0.9	0.9	0.9	1	1	1.1	1.1
2	5,333	WO	0.9	0.9	0.9	WO	0.4	0.4	0.4	0.4	0.4	0.4	0.5
3	14,040	WO	0.2	0.2	0.3	WO	0.1	0.1	0.1	0.1	0.1	0.2	0.2
4	43,661	EO	1.2	1.3	1.5	SE,EO	0.2	0.2	0.2	0.2	0.2	0.2	0.2
5	159,987	EO	0.4	0.4	0.3	SE	0.04	0.04	0.04	0.04	0.04	0.1	0.1
6	2,544	EO	36.2	36.2	48	SE	3.8	3.8	3.8	4.3	4.1	4.6	4.5
7	6,426	WO	0.6	0.6	0.6	WO	0.3	0.3	0.3	0.3	0.3	0.3	0.4
8	5,708	WO	1.3	1.5	2.7	WO	0.9	0.6	0.7	0.7	0.7	0.7	0.8
9	13,482	EO	4.5	6.2	6.7	SE	0.9	0.7	0.7	0.8	0.7	0.8	0.8
10	312,771	EO	16.8	17.8	9.1	SE,EO	0.2	0.2	0.2	0.2	0.2	0.2	0.2
11	28,703	WO	0.9	0.7	0.6	WO	0.3	0.1	0.2	0.2	0.2	0.2	0.2
12	26,761	–	∞	12.3	13.2	SE,EO	62.4	0.9	0.9	0.9	0.9	1	1
13	312,771	–	∞	12.9	24.1	–	∞	0.4	0.3	0.4	0.3	0.4	0.4
14	23,494	EO	87	90	∞	SE	4.3	4.3	2.8	4.3	3	4.3	3.3
15	57,492	WO	0.2	0.2	0.1	WO	0.1	0.04	0.04	0.04	0.04	0.04	0.1
16	23,884	WO	0.5	0.6	0.5	WO	0.3	0.1	0.1	0.2	0.2	0.2	0.2
17	2,544	–	∞	∞	∞	SE	98.6	12.6	9.6	13.1	10.5	13.7	11.3
18	122,268	SE	26.3	16.3	14.9	SE	0.6	0.4	0.3	0.4	0.4	0.4	0.4
19	101,431	WO	0.6	0.3	0.1	WO	0.5	0.04	0.04	0.04	0.04	0.04	0.04
20	149,465	WO	0.01	0.01	0.01	SE,WO	0.01	0.01	0.01	0.01	0.01	0.01	0.01
21	79,415	WO	0.3	0.2	0.2	WO	0.2	0.1	0.1	0.1	0.1	0.1	0.1
22	46,952	WO	3.9	2.1	1.9	WO	0.9	0.2	0.2	0.2	0.2	0.2	0.2
23	45,124	WO	0.7	0.7	0.7	WO	0.2	0.1	0.1	0.1	0.1	0.1	0.1
24	48,259	–	∞	∞	∞	SE	16.3	16.3	2.9	11.4	3.1	10	3.4
25	47,698	WO	0.3	0.4	0.3	WO	0.1	0.1	0.1	0.1	0.1	0.1	0.1
26	42,747	WO	1.2	1.2	0.7	SE	0.3	0.1	0.1	0.1	0.1	0.1	0.1

Table 6: Comparison of the best ballot-polling and ballot-level comparison audit methods across 26 IRV elections, with an error rate of 1% used to manipulate reported ballots. The average # of ballot samples required (expressed as a percentage of ballots cast) by the best alternate method (EO, SE, and WO) and those generated by *audit-irv* are compared. The notation ∞ indicates a percentage of ballots (or ASN) greater than 100%.

seeds used to randomly draw (sample) ballots during the audit. When reporting the ASNs and actual number of ballots sampled by each auditing method, we average these values over the 50 simulated audits. Tables 6 to 9 report the ASN and actual number of ballot samples required, on average, across the simulation of varying types of audit in each of our 26 election instances, with a 1% to 10% error rate, $\alpha = 5\%$, and $\gamma = \{1.1 \ldots 1.3\}$. We compare the EO, SE, and WO auditing methods, in both a ballot-polling and comparison audit context, against the audits generated by *audit-irv*. Tables 6 to 9 show that even when there are discrepancies between actual and reported ballots: comparison audits are still more efficient, in general, than ballot-polling audits; and *audit-irv* is able to generate efficient audits that sample only a small fraction of cast ballots.

As the rate of introduced errors increases toward 10%, the ASNs associated with the comparison audits generated by *audit-irv* significantly underestimate the actual auditing effort required in a small number of instances. This is the case in instances 10 (Pierce 2008 County Executive), 13 (Pierce 2008 County Assessor), 14 (San Leandro 2010 Mayor), 17 (Aspen 2009 City Council), 18 (Oakland 2010 Mayor) and 24 (Gosford NSW 2015). The MOV in each of these elections is less than 1% of the total ballots cast. Our results indicate that for very close elections, with a very small margin of victory, the impact of each discrepancy encountered in the sampling of ballots has a significant influence on the statistics being maintained throughout the comparison audit. Recall that the MACRO algorithm of Figure 3 repeatedly samples ballots until a running Kaplan-Markov MACRO P-value (P_{KM}) falls below the given risk limit α. When we discover a discrepancy that has resulted in the margin between a winning and losing candidate being overstated (i.e., thought to be larger than it actually is), this P_{KM} statistic increases at a rate that is proportional to the inverse of the election MOV. For elections with a very small MOV, each discovered error may significantly increase the ASN of the audit. In these instances, a full manual recount is likely to be required (and indeed, the announced outcome may be wrong).

8 Conclusion

We have presented and evaluated several methods for conducting ballot-polling and ballot-level comparison RLAs for IRV elections. These approaches represent the first practical techniques for conducting RLAs for IRV. As in FPTP, we find that comparison-based IRV audits are, in general, more efficient than their ballot-polling counterparts. These audits typically require only a small fraction of cast ballots to be sampled, though very close elections (with a MOV that is less than 1% of cast ballots, for example) generally require a full manual recount. We have presented an algorithm, denoted *audit-irv*, for designing efficient ballot-polling and ballot-level comparison-based RLAs for a given IRV election. This algorithm finds a collection of facts to audit that require the least number of expected ballot checks to confirm (assuming the announced outcome is correct), while still guaranteeing that a wrong result with be detected with probability

Auditing using audit-irv via ballot polling (BP) and comparison (CP) audits, $\alpha = 5\%$, 3% errors

#	$\lvert\mathcal{B}\rvert$	BP				CP							
		Best Alt. EO/SE/WO		*audit-irv*		Best Alt. $\gamma = 1.1$ EO/SE/WO		$\gamma = 1.1$ *audit-irv*		$\gamma = 1.2$ *audit-irv*		$\gamma = 1.3$ *audit-irv*	
		Method	Polls %	Polls %	ASN %	Method	Polls %	Polls %	ASN %	Polls %	ASN %	Polls %	ASN %
1	4,682	SE	3.7	3.8	5	SE	1	1	0.9	1.1	1	1.1	1
2	5,333	WO	0.9	0.9	0.9	WO	0.5	0.4	0.4	0.4	0.4	0.4	0.4
3	14,040	WO	0.2	0.2	0.3	WO	0.1	0.1	0.1	0.1	0.2	0.1	0.2
4	43,661	SE	1.2	1.3	1.6	SE,EO	0.2	0.2	0.2	0.2	0.2	0.2	0.2
5	159,987	SE,EO	0.4	0.4	0.3	SE	0.04	0.04	0.04	0.1	0.04	0.1	0.1
6	2,544	EO	36.2	36.2	49.7	SE	3.9	3.9	3.8	4.4	4.2	4.7	4.6
7	6,426	WO	0.6	0.6	0.7	WO	0.3	0.3	0.3	0.3	0.3	0.3	0.4
8	5,708	WO	1.3	1.5	2.8	WO	0.9	0.7	0.7	0.7	0.7	0.8	0.8
9	13,482	EO	4.5	6.2	6.9	SE	0.9	0.8	0.7	0.8	0.8	0.9	0.8
10	312,771	SE,EO	18	18.2	8.1	SE,EO	0.7	0.7	0.2	0.5	0.2	0.5	0.2
11	28,703	WO	0.9	0.7	0.6	WO	0.4	0.2	0.2	0.2	0.2	0.2	0.2
12	26,761	–	∞	12.3	12.6	EO,SE	∞	1.2	0.8	1.2	0.9	1.2	1
13	312,771	–	∞	13.2	27.2	EO,SE	∞	1	0.3	0.8	0.4	0.7	0.4
14	23,494	SE,EO	87.1	90	∞	SE	19.1	19.1	2.8	11.8	3.1	9.2	3.3
15	57,492	WO	0.2	0.2	0.1	WO	0.1	0.04	0.04	0.04	0.04	0.1	0.1
16	23,884	WO	0.5	0.6	0.5	WO	0.4	0.2	0.2	0.2	0.2	0.2	0.2
17	2,544	–	∞	∞	∞	–	∞	17.6	9.7	16.6	10.5	16.8	11.4
18	122,268	SE	16.4	16.4	13.2	SE	19.5	0.8	0.3	0.6	0.3	0.6	0.4
19	101,431	–	∞	0.3	0.1	WO	0.6	0.04	0.04	0.04	0.04	0.1	0.04
20	149,465	WO	0.01	0.01	0.01	SE,WO	0.01	0.01	0.01	0.01	0.01	0.01	0.01
21	79,415	WO	0.3	0.2	0.2	WO	0.2	0.1	0.1	0.1	0.1	0.1	0.1
22	46,952	WO	4.3	2.1	1.9	WO	1.2	0.2	0.2	0.2	0.2	0.2	0.2
23	45,124	SE,WO	0.7	0.7	0.7	SE	0.1	0.2	0.2	0.1	0.1	0.1	0.1
24	48,259	–	∞	∞	∞	SE	76.8	76.8	2.3	69	2.5	62.3	2.7
25	47,698	WO	0.3	0.4	0.4	WO	0.2	0.1	0.1	0.1	0.1	0.1	0.1
26	42,747	WO	1.2	1.2	0.7	WO	0.4	0.1	0.1	0.1	0.1	0.1	0.1

Table 7: Comparison of the best ballot-polling and ballot-level comparison audit methods across 26 IRV elections, with an error rate of 3% used to manipulate reported ballots. The average # of ballot samples required (expressed as a percentage of ballots cast) by the best alternate method (EO, SE, and WO) and those generated by *audit-irv* are compared. The notation ∞ indicates a percentage of ballots (or ASN) greater than 100%.

Auditing using audit-irv via ballot polling (BP) and comparison (CP) audits, $\alpha = 5\%$, 5% errors

#	$\lvert\mathcal{B}\rvert$	BP				CP							
		Best Alt. EO/SE/WO		*audit-irv*		Best Alt. $\gamma = 1.1$ EO/SE/WO		$\gamma = 1.1$ *audit-irv*		$\gamma = 1.2$ *audit-irv*		$\gamma = 1.3$ *audit-irv*	
		Method	Polls %	Polls %	ASN %	Method	Polls %	Polls %	ASN %	Polls %	ASN %	Polls %	ASN %
1	4,682	SE,EO	3.7	3.7	5	SE	1.1	1.1	1	1.1	1	1.2	1.1
2	5,333	WO	0.9	0.9	1	WO	0.5	0.4	0.4	0.4	0.4	0.4	0.5
3	14,040	WO	0.2	0.2	0.3	WO	0.1	0.1	0.1	0.1	0.2	0.2	0.2
4	43,661	SE,EO	1.3	1.3	1.7	EO	0.3	0.2	0.2	0.2	0.2	0.3	0.2
5	159,987	SE,EO	0.4	0.4	0.3	SE	0.1	0.1	0.04	0.1	0.04	0.1	0.1
6	2,544	EO	36.3	36.3	50.7	SE	5.5	5.5	3.9	5.4	4.3	5.6	4.6
7	6,426	WO	0.6	0.6	0.7	WO	0.3	0.3	0.3	0.3	0.3	0.3	0.4
8	5,708	WO	1.4	1.5	2.9	WO	1	0.7	0.7	0.8	0.7	0.8	0.8
9	13,482	EO	4.5	6.2	7	SE	1.1	0.9	0.7	0.9	0.8	0.9	0.8
10	312,771	SE,EO	18.5	18.5	7.1	SE,EO	18.8	19.7	0.1	1.4	0.2	0.7	0.2
11	28,703	WO	0.9	0.7	0.6	WO	0.4	0.2	0.2	0.2	0.2	0.2	0.2
12	26,761	–	∞	12.3	13.2	–	∞	1.3	0.9	1.3	0.9	1.3	1
13	312,771	–	∞	13.3	30	–	∞	27.1	0.3	2.7	0.4	1.4	0.4
14	23,494	SE	86.3	89.5	∞	SE	58.8	58.8	2.6	40.5	2.8	26.4	3
15	57,492	WO	0.2	0.2	0.1	WO	0.1	0.04	0.04	0.1	0.1	0.1	0.1
16	23,884	WO	0.5	0.7	0.5	WO	0.4	0.2	0.2	0.2	0.2	0.2	0.2
17	2,544	–	∞	∞	∞	–	∞	22.1	10.2	19.6	11.1	19.3	12.1
18	122,268	SE	25.7	16.5	12.1	–	∞	2.6	0.3	1.1	0.3	0.9	0.4
19	101,431	WO	0.7	0.3	0.1	WO	0.7	0.04	0.04	0.04	0.04	0.1	0.04
20	149,465	WO	0.01	0.01	0.01	SE,WO	0.01	0.01	0.01	0.01	0.01	0.01	0.01
21	79,415	WO	0.3	0.2	0.2	WO	0.2	0.1	0.1	0.1	0.1	0.1	0.1
22	46,952	SE	3.2	2.1	1.9	SE	0.3	0.3	0.2	0.3	0.2	0.3	0.2
23	45,124	SE	0.7	0.7	0.7	SE	0.1	0.1	0.1	0.1	0.1	0.1	0.1
24	48,259	SE	97.9	97.9	∞	SE	90.1	90.1	1.9	92	2	87	2.2
25	47,698	WO	0.3	0.4	0.4	WO	0.2	0.08	0.08	0.1	0.1	0.1	0.1
26	42,747	WO	1.2	1.2	0.7	WO	0.5	0.1	0.1	0.1	0.1	0.2	0.1

Table 8: Comparison of the best ballot-polling and ballot-level comparison audit methods across 26 IRV elections, with an error rate of 5% used to manipulate reported ballots. The average # of ballot samples required (expressed as a percentage of ballots cast) by the best alternate method (EO, SE, and WO) and those generated by *audit-irv* are compared. The notation ∞ indicates a percentage of ballots (or ASN) greater than 100%.

Auditing using audit-irv via ballot polling (BP) and comparison (CP) audits, $\alpha = 10\%$, 10% errors

#	$\lvert\mathcal{B}\rvert$	BP				CP									
		Best Alt. EO/SE/WO		*audit-irv*		Best Alt. $\gamma = 1.1$ EO/SE/WO		$\gamma = 1.1$ *audit-irv*		$\gamma = 1.2$ *audit-irv*		$\gamma = 1.3$ *audit-irv*	
		Method	Polls %	Polls %	ASN %	Method	Polls %	Polls %	ASN %	Polls %	ASN %	Polls %	ASN %
1	4,682	SE,EO	3.7	3.8	5.4	SE	1.3	1.2	1	1.2	1.1	1.3	1.2
2	5,333	WO	0.9	0.9	1.1	WO	0.6	0.5	0.4	0.5	0.5	0.5	0.5
3	14,040	WO	0.3	0.2	0.3	WO	0.2	0.1	0.1	0.1	0.2	0.2	0.2
4	43,661	SE,EO	1.3	1.3	2	EO	0.4	0.4	0.2	0.4	0.2	0.3	0.2
5	159,987	SE,EO	0.4	0.4	0.4	SE	0.1	0.1	0.04	0.1	0.1	0.1	0.1
6	2,544	EO	37.2	37.2	60.8	SE	6.2	6.2	4.3	5.8	4.7	5.9	5.1
7	6,426	WO	0.6	0.6	0.7	WO	0.4	0.3	0.3	0.3	0.3	0.4	0.4
8	5,708	WO	1.7	1.5	3.1	WO	1.1	0.7	0.7	0.8	0.8	0.8	0.8
9	13,482	EO	4.6	6.2	7.4	EO	1.8	1.2	0.7	1.2	0.8	1.1	0.9
10	312,771	SE,EO	20.7	20.8	5.4	SE,EO	78.1	90	0.12	78.1	0.1	74.1	0.1
11	28,703	WO	1	0.7	0.7	WO	0.6	0.2	0.2	0.2	0.2	0.23	0.2
12	26,761	–	∞	12.3	12.8	–	∞	2.3	0.9	1.7	0.9	1.6	1
13	312,771	–	∞	14.3	39	–	∞	80.2	0.4	53.9	0.4	50.7	0.5
14	23,494	SE	86.2	88.9	∞	SE	89	86.2	2.6	84.4	2.8	84.4	3
15	57,492	WO	0.2	0.2	0.1	WO	0.1	0.1	0.04	0.1	0.1	0.1	0.1
16	23,884	WO	0.5	1	0.6	WO	0.4	0.2	0.2	0.2	0.2	0.2	0.2
17	2,544	–	∞	∞	∞	–	∞	44.1	9.5	35.2	10.4	31.6	11.2
18	122,268	SE	14.1	16.5	10.1	SE	92.1	66.7	0.3	29.2	0.3	9	0.3
19	101,431	WO	0.9	0.3	0.1	WO	17.7	0.1	0.04	0.1	0.04	0.1	0.04
20	149,465	WO	0.01	0.01	0.01	SE,WO	0.01	0.01	0.01	0.01	0.01	0.01	0.01
21	79,415	WO	0.3	0.2	0.2	WO	0.3	0.1	0.1	0.1	0.1	0.1	0.1
22	46,952	SE	3.2	2.1	2	SE	0.5	0.4	0.2	0.3	0.2	0.3	0.2
23	45,124	SE	0.7	0.7	0.7	SE	0.1	0.2	0.1	0.1	0.1	0.1	0.1
24	48,259	SE	84.8	84.8	∞	SE	98	98	1.4	98	1.5	96	1.7
25	47,698	WO	0.4	0.3	0.4	WO	0.2	0.1	0.1	0.1	0.1	0.1	0.1
26	42,747	WO	1.3	1.2	0.7	WO	0.7	0.2	0.1	0.2	0.1	0.2	0.1

Table 9: Comparison of the best ballot-polling and ballot-level comparison audit methods across 26 IRV elections, with an error rate of 10% used to manipulate reported ballots. The average # of ballot samples required (expressed as a percentage of ballots cast) by the best alternate method (EO, SE, and WO) and those generated by *audit-irv* are compared. The notation ∞ indicates a percentage of ballots (or ASN) greater than 100%.

at least $1 - \alpha$. The audit configurations generated with this algorithm are competitive with alternate methods considered throughout the paper, and in some cases are substantially more efficient.

References

References

[1] T. Antonyan, S. Davtyan, S. Kentros, A. Kiayias, L. Michel, N. Nicolaou, A. Russell, and A. A. Shvartsman. State-wide elections, optical scan voting systems, and the pursuit of integrity. *IEEE Transactions on Information Forensics and Security*, 4(4):597–610, 2009.

[2] B. Beckert, M. Kirsten, V. Klebanov, and C. Schürmann. Automatic margin computation for risk-limiting audits. In *International Joint Conference on Electronic Voting*, pages 18–35. Springer, 2016.

[3] J. Benaloh, D. Jones, E. Lazarus, M. Lindeman, and P.B. Stark. Soba: Secrecy-preserving observable ballot-level audit. In *USENIX Accurate Electronic Voting Technology Workshop*, 2011.

[4] M. Blom, P. J. Stuckey, V. Teague, and R. Tidhar. Efficient Computation of Exact IRV Margins. In *European Conference on AI (ECAI)*, pages 480–487, 2016.

[5] S. Checkoway, A. D. Sarwate, and H. Shacham. Single-ballot risk-limiting audits using convex optimization. In *EVT/WOTE*, 2010.

[6] J.L. Hall, L.W. Miratrix, P.B. Stark, M. Briones, E. Ginnold, F. Oakley, M. Peaden, G. Pellerin, T. Stanionis, and T. Webber. Implementing risk-limiting post-election audits in California. In *Proc. 2009 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE '09)*, Montreal, Canada, August 2009. USENIX.

[7] M. Lindeman and P.B. Stark. A gentle introduction to risk-limiting audits. *IEEE Security and Privacy*, 10:42–49, 2012.

[8] M. Lindeman, P.B. Stark, and V. Yates. BRAVO: Ballot-polling risk-limiting audits to verify outcomes. In *Proceedings of the 2011 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE '11)*. USENIX, 2012.

[9] Mark Lindeman. Evidence-based elections: Beyond the rigging debate. *Significance*, 14(1):18–23, 2017.

[10] Mark Lindeman, Neal McBurnett, Kellie Ottoboni, and Philip B Stark. Next steps for the colorado risk-limiting audit (corla) program. *arXiv preprint arXiv:1803.00698*, 2018.

[11] T.R. Magrino, R.L. Rivest, E. Shen, and D.A. Wagner. Computing the margin of victory in IRV elections. In *USENIX Accurate Electronic Voting Technology Workshop: Workshop on Trustworthy Elections*, USENIX Association Berkeley, CA, USA, 2011.

[12] Lawrence D Norden and Ian Vandewalker. *Securing Elections from Foreign Interference*. Brennan Center for Justice at the New York University School of Law, 2017.

[13] Ronald L Rivest and Philip B Stark. When is an election verifiable? *IEEE Security & Privacy*, 15(3):48–50, 2017.

[14] A.D. Sarwate, S. Checkoway, and H. Shacham. Risk-limiting audits and the margin of victory in nonplurality elections. *Politics, and Policy*, 3(3):29–64, 2013.

[15] P.B. Stark. Risk-limiting post-election audits: P-values from common probability inequalities. *IEEE Transactions on Information Forensics and Security*, 4:1005–1014, 2009.

[16] P.B. Stark and M. Lindeman. A gentle introduction to risk-limiting audits. *IEEE SECURITY and PRIVACY*, 10:42–49, 2012.

[17] Philip B Stark. Auditing a collection of races simultaneously. *arXiv preprint arXiv:0905.1422*, 2009.

[18] Philip B Stark. Efficient post-election audits of multiple contests: 2009 california tests. 2009.

[19] Philip B Stark. Super-simple simultaneous single-ballot risk-limiting audits. In *EVT/WOTE*, 2010.

[20] Philip B Stark and Vanessa Teague. Verifiable european elections: Risklimiting audits for dhondt and its relatives. *USENIX Journal of Election Technology and Systems (JETS)*, 1(3):18–39, 2014.