Spanning Trees and Redistricting: New Methods for Sampling and Validation

Sarah Cannon,[1] Moon Duchin,[2*] Dana Randall,[3] Parker Rule[4]

[1]Department of Mathematical Sciences, Claremont McKenna College, Claremont, CA 91711
[2]Data Science Institute, University of Chicago, Chicago, IL 60615
[3]School of Computer Science, Georgia Institute of Technology, Atlanta, GA 30332
[4]Tisch College of Civic Life, Tufts University, Medford, MA 02155 USA

*To whom correspondence should be addressed; E-mail: mduchin@uchicago.edu.

November 16, 2025

Abstract

Deciding whether a political districting plan was distorted by a hidden agenda, or whether it dilutes the voting power of some group, requires a neutral baseline for comparison. Remarkably, all nine U.S. Supreme Court justices have now signed on to decisions that find that computational methods can provide key evidence. Today, the leading approaches for benchmarking districting plans are based on the use of spanning trees for sampling graph partitions. We present a new *reversible recombination* algorithm and rigorously prove its fundamental properties. Furthermore, we argue for a canonical sampling distribution called the *spanning tree distribution* that is well adapted to redistricting and provides a principled foundation for comparing and validating methods. Together with a highly efficient (and open-source) implementation that can generate and handle large datasets, this work provides the most powerful null model to date for the gerrymandering problem, meeting an urgent democratic challenge with sound scientific methodology.

1 Introduction

1.1 Redistricting

Throughout the world, many countries divide their territory into regions that each conduct legislative elections, from the provinces of South Africa to the departments of France to the states of Brazil. Any update to the boundary lines—*redistricting*—can have a significant impact on representational outcomes. Gerrymandering, the practice of abusing line-drawing power by boosting representation for a favored group over other priorities, is constantly in American news—and courtrooms.[1] Yet for many decades, the U.S. Supreme Court has struggled to identify the non-gerrymandered baseline: How much representation should groups expect from a "neutral" redistricting process?

Starting around 2013, experts working in U.S. courts of law have presented computational techniques to sample from the (very large) space of plausible districting plans, arguing that a random *ensemble* of plans generated without sensitive data (such as partisan or racial data) provides a neutral statistical baseline. The underlying idea of comparing a proposed plan to a collection of alternative plans to diagnose the principles of its design is called the *ensemble method* for redistricting analysis.

[1]It is a singularly acute problem in the United States because lines must be regularly redrawn, and usually by elected officials themselves. And the constant high-profile cases come because redistricting cases are among the only ones that still receive mandatory Supreme Court review (28 U.S.C. §1253).

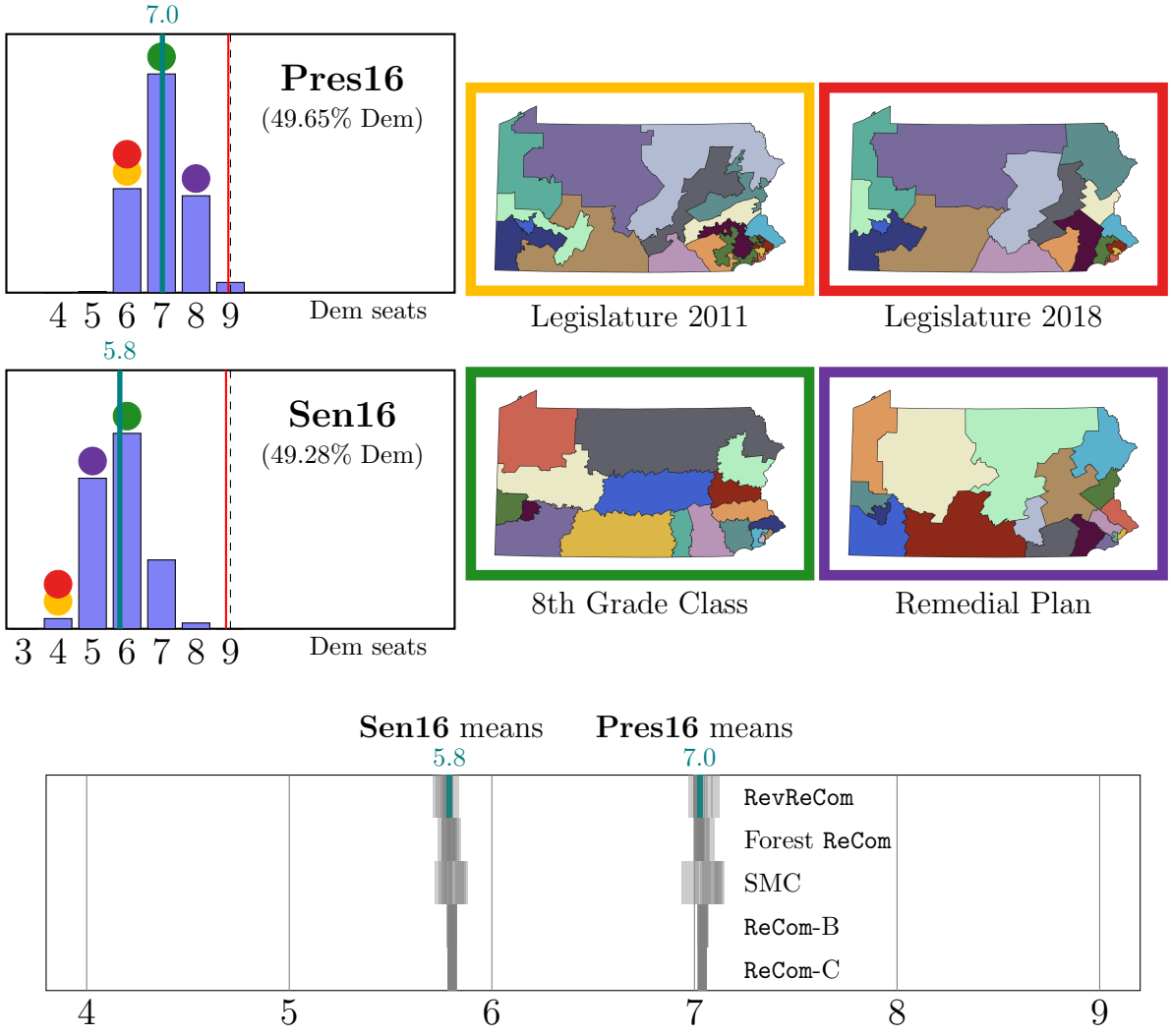


Figure 1: **Illustration of the ensemble method in Pennsylvania.** TOP LEFT: An ensemble of Pennsylvania districting plans produced with `RevReCom` is shown in the histograms (in blue). To make it, we overlay the plans with voting from two elections to determine how many of the 18 districts have more D votes than R votes; we call these Democratic seats. These "blindly drawn" plans usually gives fewer Democratic seats than the proportional outcome of nearly nine—an empirical finding based on the detailed geography of votes. TOP RIGHT: Four Congressional plans can then be compared to the ensemble, with colored dots on the histograms marking their performance. The (Republican) legislature's plans both secure more (Republican) partisan advantage than the bulk of the ensemble. BOTTOM: To investigate the consistency of tree-based sampling methods, we repeat this process and plot the mean of each trial. We include ten independent ensembles made with `RevReCom` and with four other samplers and plot the means with light gray bars, which appear darker when they overlap. (The means from the histograms above are shown here in teal.) The methods with asymptotic distributional guarantees (`RevReCom`, Forest `ReCom`, and SMC) require more computation and give slightly more variable results, while the heuristic `ReCom` variants (B,C) give fast and stable results, but come with only an approximate description for their target distribution. See github for details.

Ensemble evidence has been used in court cases in at least 11 states since the 2020 Census, and the number continues to grow in the mid-decade redistricting sprint. Its wide use in redistricting litigation has led to a series of Supreme Court decisions in which *all nine* sitting justices have signed on to opinions describing this kind of evidence as potentially useful for their voting rights work.[2] It is remarkable for a famously math-skeptical Supreme Court to cite a class of graph algorithms as providing useful evidence to address a half-century-old puzzle in their jurisprudence.

We begin with a motivating example. Figure 1 shows the ensemble method in practice, illustrating issues that arose in *League of Women Voters v. Pennsylvania* in 2018. We first demonstrate the method using an ensemble made with `RevReCom`, the algorithm introduced here, to randomly draw a large number of districting plans that satisfy population balance, contiguity, and compactness without any partisan data or objectives. Pairing results from any given election with a given set of districts lets us calculate how many seats each political party would win in that vote pattern; these values, across all plans, form histograms like those shown in Figure 1. Four plans in circulation at the time of the lawsuit are depicted in the figure, with their partisan performance marked in the histograms. This constructed baseline is precisely what we need to disambiguate geography from gerrymandering in the proposed plans.

These two vote patterns illustrate realistic ways that Pennsylvanians had expressed partisan preference in contemporaneous elections. Interestingly, although those two elections each had a nearly 50–50 vote split between the major parties, the parties do not tend to get a similar seat split in the random plans: in the 2016 Presidential vote pattern, Democrats can anticipate controlling 7.0 districts out of 18 in expectation (under 39%), while the 2016 U.S. Senate vote pattern gives an expectation of just 5.8 seats (roughly 32%). Thus, the use of an ensemble of alternatives lets us measure the effect size of the partisan advantage created by the geography of the voters and the rules of redistricting. This gap of over a seat in expectation between the two elections is in no way visible by eyeballing a standard map of the voting pattern; it reflects subtle differences in the geography that require an empirical method to draw out.[3] For further discussion of the ways that specific geographic configurations can impact representation, see also [Duc22, RW22].

The Legislature's enacted plan from 2011 (red) and the much more "compact" proposal from 2018 (yellow) look different to the eye, but they perform similarly against both vote patterns, giving a Republican advantage that pushes beyond the lean of party-neutral plans. The court's remedial plan (purple) is highly responsive to changes in the vote, swinging by three seats under the subtle shift between the two elections. And finally, the plan drawn by an 8th grade class (green) sits at the highest bar both times, behaving just as though it was drawn with no partisan data—which it was.

Figure 1 also lets us compare the seat share estimate from `RevReCom` to values obtained from other methods (introduced below in §2.2) run at their largest practical sample sizes; we see that a variety of tree-based methods considered in this paper, set to approximately target the same distribution on plans, all produce similar quantitative conclusions about partisan expectation.

1.2 Graphs, partitions, and Markov chains

Sampling balanced partitions of a set can be challenging computationally, especially under geometric constraints. In redistricting, not only must pieces be connected and population-balanced, but it is also preferable to have compact (nicely shaped) components rather than elongated, spindly, or contorted ones.[4] To make this mathematically precise, we represent the region by a weighted graph, with nodes

[2] The opinion, concurrences, and dissents in *Rucho v. Common Cause* (2019) and *Allen v. Milligan* (2023) contain statements that cover all nine justices—see §A for more details.

[3] In those elections, Donald Trump outpolled Hillary Clinton 2,970,733–2,926,441 in the Presidential contest (Pres16) while the Pat Toomey advantage over Katie McGinty was 2,951,702–2,865,012 in the U.S. Senate race (Sen16)—a nearly equal number of total votes, and a nearly equal split. That means that the difference is not attributable to "rolloff" (where one contest had substantially more votes cast) or to third-party candidates, but legitimately reflects the geographical distribution of Clinton voters vs. McGinty voters. The analysis reveals, in particular, that there were significant numbers of ticket-splitters (voting R in one contest and D in the other) in both directions.

[4] For contorted districts, compare the original gerrymander: `https://www.masshist.org/database/1765`.

for basic indivisible geographical units and edges between nodes when the corresponding units are adjacent. The units—such as census blocks or precincts—are weighted by population.

In this formulation, a *districting plan* is a partition of the graph into k connected subgraphs with nearly equal population. Building a neutral (non-gerrymandered) baseline for redistricting purposes can be accomplished with an appropriate random sample from the set of balanced graph partitions. Setting this up as a graph problem lends itself to a particularly simple measure of compactness via the size of the boundary.

Many methods have been introduced for randomly generating districting plans; for a 60-year overview, see [BS22]. Here, our main focus will be on Markov chain Monte Carlo (MCMC) algorithms, which are ubiquitous across scientific disciplines for the randomized study of large, complicated configuration spaces (for surveys, see [Dia09, AF02, Fel68]). The idea is to design a random walk that moves among states in a state space—in this case, stepping from partition to partition. Though each plan might only have a few neighbors, MCMC seeks to run until convergence to a useful *stationary distribution* (or *steady state*) over the entire state space; that is, the probabilities of being at each state will stabilize, so that further steps maintain the probability distribution.

1.3 Goals

To create the benchmark for redistricting that the courts have sought, we aim to generate *representative samples* of partitions. This is notably distinct from the frequent use of MCMC for *optimization*, where one targets a stationary distribution weighted towards "better" configurations, seeking local or global optima. It is also distinct from *exploration*, as in generating diverse instances of configurations of an Ising model or spin glass. In contrast to these goals, we will designate a meaningful weighting of plans corresponding to a core set of real-world rules and priorities—such as a preference for nicer shapes—and then target this distribution. Our ensembles will be built by collecting each plan visited by the random walk; after enough steps, the statistics of the ensemble should approximate draws from the chain's stationary distribution.[5] This gives us a notion of typical or expected properties, given a set of rules and priorities.

In this article, we introduce a new *reversible recombination* Markov chain (`RevReCom`) that provably converges to the precise *spanning tree distribution* that various heuristics and algorithms have been built to target.[6] We argue below that this probability measure based on spanning tree counts is an excellent choice of reference distribution for sampling compact graph partitions, well suited to meeting the needs of courts and policymakers. Finally, we leverage our highly efficient software implementation to expand the repertoire of benchmarking techniques in the literature, offering numerous comparisons of spanning-tree-based samplers and highlighting caveats and limitations. An overarching finding is that, when they are run with adequate sample sizes and clean convergence diagnostics, tree-based samplers provide a growing suite of generally reliable tools.

2 Partitioning with Spanning Trees

2.1 Spanning trees and community structure

A vast number of applications in theoretical computer science and engineering rely on the use of *spanning trees* of a graph G, which are cycle-free subgraphs of G using all vertices. They contain the minimum amount of connective tissue to keep all the nodes in one component; that is, deleting any edge cuts the graph into (exactly) two pieces. The number of spanning trees $N_{\mathsf{ST}}(G)$ of a graph G, sometimes referred to as its *complexity*, is a measure of richness or well-connectedness that is efficiently

[5] In many applications, researchers will employ parameters s_1 and s_2 to implement *burn-in* and *sub-sampling*: a Markov chain process will skip the first s_1 states before adding a state to the ensemble; subsequently, every s_2^{th} state visited by the chain will be added. For more discussion of burn-in and sub-sampling, as well as sample size, see §B.2.

[6] We achieve this with combinatorially simple rejection steps rather than a Metropolis filter as in [ACH+23]; we will offer empirical comparisons of the approaches, which give mutually reinforcing results.

computed using graph Laplacians [Lyo05]. A simple graph on n nodes can have a spanning tree count anywhere from 1 (if it is itself a tree) to n^{n-2} (if all possible edges are present); the number grows quickly as the graph gets larger and more internally connected. Figure 2 highlights two pieces or "districts" in partitions P and Q of a grid-graph. One highlighted district has 15 spanning trees and the other has just 1 spanning tree (because it is a tree itself). We can take the size of the cut-set of a partition (the number of edges with their endpoints in different parts, also called the number of *cut edges*) to be a measure of the efficiency of the partition.

Small cut-sets, high internal connectivity within districts, and efficient-looking shapes are all simultaneously achieved when the pieces of a graph partition have relatively many spanning trees [DT24]. Accordingly, a natural choice of weighting for a graph partition is to use the product of the spanning tree counts of its districts—in other words, if P is a partition of a graph into connected subgraphs $P_1, \ldots, P_k$ representing its districts, then we want to sample according to $N_{\mathsf{ST}}(P) := \prod_i N_{\mathsf{ST}}(P_i)$. We thus define a weight proportional to this spanning tree count,

$$\pi(P) \propto N_{\mathsf{ST}}(P),$$

and call this the *spanning tree distribution* on partitions. It is intuitively clear that weighting by π favors "plump" districts, because those typically have more spanning trees than more "spindly" or tree-like districts do, as illustrated in Figure 2.

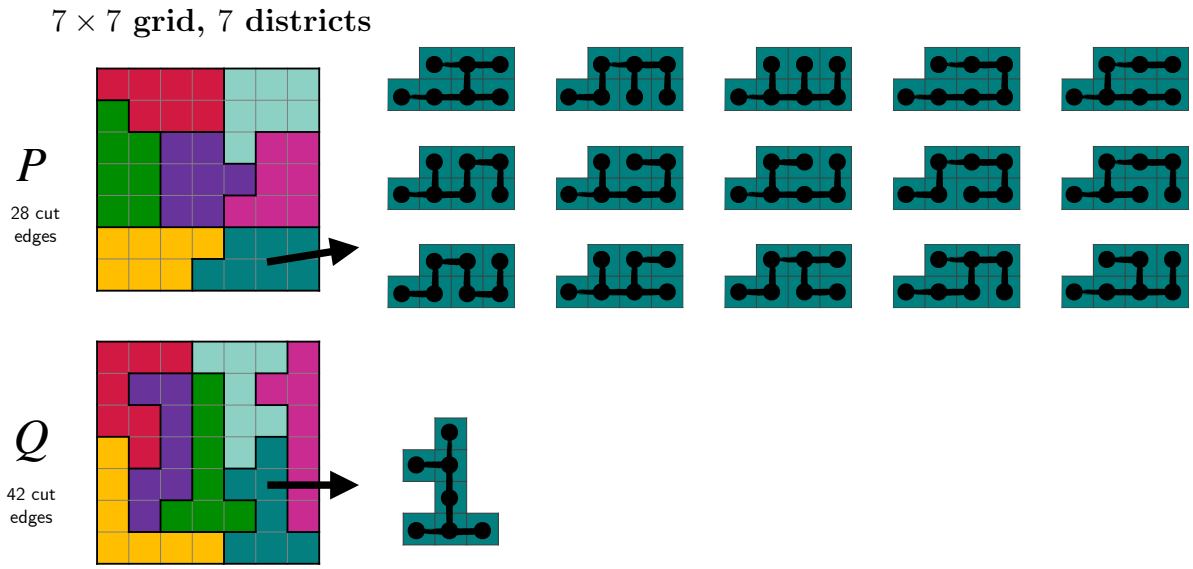


Figure 2: **Weighting partitions with spanning trees.** Two configurations or districting plans P and Q are shown here, for the $7 \times 7 \to 7$ districting problem. There are 28 cut edges in plan P and 42 cut edges in plan Q. In this example, each district Q_i in Q has just one spanning tree, while each district P_i in P has 15 spanning trees. This means a sample from the spanning tree distribution π is exactly 15^7 times as likely to choose the "plump" plan P as it is to choose the "spindly" plan Q—a factor of over 170 million.

For the problem of dividing a 7×7 grid into 7 districts of equal size, partition P from Figure 2 is 15^7 times more likely to appear as a sample under this distribution than partition Q is. This very heavy preference is balanced out by the fact that there are many more plans with long boundaries than there are highly compact plans. Across all partitions of the 7×7 grid, Table 1 confirms the overall impact of re-weighting by spanning tree count: more compact partitions with shorter boundaries are favored in this distribution, bringing the average size of a cut-set down from roughly 37.6 to 31.9.

cut edges	number of plans (exact count)	total spanning tree weight (millions, rounded)
28	420	73447
29	5408	250666
30	43468	528671
31	219704	698394
32	884620	732191
33	2686928	577890
34	6578950	366429
35	12985744	186993
36	21167576	78541
37	28289752	26977
38	31084950	7589
39	27036848	1684
40	17848860	282
41	7971064	32
42	1949522	2

Table 1: **Spanning tree weight factors.** This table quantifies by how much the spanning tree distribution up-weights more compact plans (those with a smaller cut-set) relative to the complete enumeration—the center column is uniform, while the right column is formed by summing the spanning tree weight $N_{\mathsf{ST}}(P)$ over all plans with each number of cut edges. The L^1 Wasserstein distance between distributions can be calculated after normalizing each to total mass one, obtaining $d_{\text{Wass}}(\text{uniform}, \pi) = 5.72785$. The average cut edge count shifts correspondingly, from roughly 37.61 (when all plans are equally weighted) to 31.88 (when we sample from the distribution π). This represents a marked improvement in compactness.

Conceptually, the spanning tree count reflects how effectively the plan picks out strongly internally connected districts in a geographical network, under the additional constraint of balanced population. Selecting highly connected subgraphs has an ample literature of its own, going by the name *community detection* in the network science literature [POM09, Moo17]. The explicit use of spanning tree counts for community detection is employed in numerous papers, including [KW08]. Whether this application of clustering might correspond well to *social* understandings of community is an interesting question that was broached by Nelson in [Nel22], and is worth serious attention in future work.

There is a growing body of work exploring more precise relationships between spanning tree counts and cut-sets. Clelland et al. investigate, for $k = 2$ districts, the strong (negative) linear relationship between $|\partial P|$ (the cut edge count) and the modified spanning tree weight $\log(N_{\mathsf{ST}}(P)|\partial P|)$, in both grids and real-world examples [CBH$^+$21]. A subsequent paper by Procaccia and Tucker-Foltz [PTF22] uses effective resistance to prove an asymptotically exponential relationship between the cut edge count and the spanning tree count. The constants in this exponential relationship can vary depending on the structure of the graph. In the special case of grid-graphs, the strongest known results are those given by Tapp [Tap24].

If working within the statistical physics paradigm, one may be tempted to try to sample plans with small values for $|\partial P|$ more directly by targeting a distribution proportional to $e^{-\beta|\partial P|}$. One downside of this approach is the added complication/discretion of choosing the value of β. Another issue is that known results show that sampling from this distribution is NP-hard in some planar graphs [NDS].

2.2 Tree-based sampling methods

Earlier research on redistricting with Markov chains focused on local or quasi-local moves that change the district assignment of one or a small number of units; these are generally called *flip steps* because they can be visualized as flipping the 'spin' or color of the nodes, analogous to Glauber dynamics

in statistical physics [BS22]. The difficulties of sampling approximately balanced partitions in a flip chain are well known, even on simple grid graphs, as flip chains face extremely slow convergence (similar to low-temperature models in statistical physics) [DDS21, FP23, NDS]. In other words, these chains have prohibitively long *mixing times*—they require many steps to pass a threshold of closeness to the stationary distribution. Often, allowing large, non-local perturbations in a single move can significantly speed up a Markov chain, as in various classic card-shuffling examples.[7]

A family of large-step Markov chains called *recombination* (or `ReCom`) was introduced by DeFord, Duchin, Najt, and Solomon [DDS21, NDS, NDS21]. The basic recombination step proposes two districts to be fused, then selects a random spanning tree of the fused double-district, then (if possible) chooses an edge from that spanning tree whose deletion leaves two components with population balance within the prescribed tolerance, thereby defining two new districts.

There are several natural variants within the `ReCom` family, based on how the choices of adjacent districts and spanning trees are randomized. One choice is whether to select the pair of districts to merge by randomizing indices or by taking the districts spanned by a random cut edge of the partition; a second choice is whether to generate a random spanning tree uniformly (UST), or by randomizing edge weights and taking the minimum-weight spanning tree (MST). In the comparisons that follow, we will designate these as `ReCom`-A (cut edge, MST), `ReCom`-B (district index, MST), `ReCom`-C (cut edge, UST), and `ReCom`-D (district index, UST). Surprisingly, while each of these `ReCom` variants approximately targets the spanning tree distribution, no closed-form representation of their precise steady state is known when there are more than two districts. We compare these variants empirically below.

These ideas—leveraging spanning trees for graph partitioning, and targeting the spanning tree distribution π described here—have inspired multiple related samplers. Mattingly et al. introduced a Metropolis-Hastings variant called Forest `ReCom` [ACH$^+$23, ACH$^+$21]. Without further tuning (i.e., with parameters $\gamma = \beta = 0$ in their implementation), the resulting process targets π. Later, Charikar et al. proposed an alternative method to target π in [CLLV22], incorporating some additional balance factors. Together with a recent result of Cannon–Pegden–Tucker-Foltz that confirms that a polynomial fraction of trees admit a balanced cut in grid-like graphs [CPTF24], the Charikar team's method runs in polynomial time—but still slowly in practice. Abrishami et al. have implemented a direct-sampling method to cut a single tree $k-1$ times and obtain a π-distributed partition.[8] And McCartan–Imai have developed an importance sampling method called sequential Monte Carlo, or SMC, that works with many copies of the initial graph and draws spanning trees to mark additional districts in a multi-stage process [MI23]. Here too, the default target is π.[9] New methods have been accompanied by a recent string of theoretical advances regarding properties of recombination chains [AKK$^+$22, PTF22, FP23, Can24].

Here, we introduce a simple modification to `ReCom` that makes it *reversible*, letting us show that the steady state is precisely the spanning tree distribution on plans. This is achieved by adding more rejection conditions to make the chain *lazier*, or more likely to self-loop, putting higher multiplicity on certain states in a controlled manner. We compensate for the high rejection rates with a parallelization strategy described in §C.1, maintaining high efficiency overall. With its rigorous foundation together with a fast implementation and customized data pipeline and compression scheme, `RevReCom` lets us conduct large experiments and make extensive comparisons between many of the spanning tree methods used in the field. Our primary experimental comparisons will put `RevReCom` alongside Forest `ReCom`, SMC, and the original `ReCom` variants A,B,C,D. Throughout the paper, we ran each method at its *largest practical sample size*—this means we used the authors' latest software implementations with uniform resource constraints of time (48 hours of runtime), memory (100GB RAM), and storage (2GB after efficient compression).

[7] Using riffle-type moves that affect the whole deck produces significantly faster mixing than iterating single-card moves—the mixing time drops to the order of $\log n$ instead of $n \log n$ for decks of n cards [AD86, DS81, Dia88, BD92].

[8] Code can be found here and here.

[9] This method is quite popular among political scientists. Interestingly, part of the reason is that it is coded in R rather than Python.

3 Reversible ReCom

3.1 Exact balance

We first define reversible recombination (`RevReCom`) for the simple case where nodes have equal weight and graph partitions require exact balance; we relax these assumptions in Section 3.2. Let $\mathcal{P}_k(G)$ be the set of connected k-partitions on a graph G with an equal division of nodes. Let $P_1, \ldots, P_k$ denote the k subgraphs induced by the pieces of the vertex partition, which we will call *districts*.[10] We call an edge e of a tree T a *balance edge* if its two complementary components have the same number of nodes. For subgraphs A, B of G, let $E(A, B)$ be the set of edges in G with one endpoint in $V(A)$ and one in $V(B)$. Let $N_{\mathsf{ST}}(A, B)$ be the number of spanning trees of the induced graph on $V(A) \cup V(B)$ that have exactly one edge in $E(A, B)$; since such a tree consists of a spanning tree on each of A and B plus that one edge,

$$N_{\mathsf{ST}}(A, B) = N_{\mathsf{ST}}(A) \cdot N_{\mathsf{ST}}(B) \cdot |E(A, B)|.$$

Then the spanning tree distribution π on $\mathcal{P}_k(G)$ is defined by

$$\pi(P) := \frac{\prod_{i=1}^{k} N_{\mathsf{ST}}(P_i)}{Z},$$

where $Z = \sum_{R \in \mathcal{P}_k(G)} \prod_{i=1}^{k} N_{\mathsf{ST}}(R_i)$ is the *normalizing constant* ensuring π is a distribution.[11]

We define the Markov chain proposal as follows. From a plan P, the fusion step is made according to district indices: first choose pairs of indices uniformly from $\{1, \ldots, k\}^2$ and let $A = P_i$ and $B = P_j$. If A and B are not adjacent and distinct, reject. If they are adjacent and distinct, consider the graph $A \cup B$ induced by G on the vertex set $V(A) \cup V(B)$ and choose a spanning tree uniformly at random. If there is no balance edge, reject. If there is, then it is unique; delete the balance edge and let the new components be called A', B' and the corresponding new plan be called Q. Accept Q with probability $1/|E(A', B')|$; else, reject. If the acceptance condition was not met, resample a new pair of indices and repeat.

Note that the number of spanning trees for $A \cup B$ that could have produced districts A' and B' is exactly $N_{\mathsf{ST}}(A', B')$. The process above prescribes a transition probability $X_P(Q)$ for transitioning from state P to state Q in a single step, as follows:

$$X_P(Q) = \frac{2}{k^2} \cdot \frac{1}{N_{\mathsf{ST}}(A \cup B)} \cdot \frac{1}{E(A', B')} \cdot N_{\mathsf{ST}}(A', B') = \frac{2}{k^2} \cdot \frac{N_{\mathsf{ST}}(A')\, N_{\mathsf{ST}}(B')}{N_{\mathsf{ST}}(A \cup B)}$$

if suitable A, B, A', B' exist (in which case they are determined up to relabeling), and zero if not. Noting that $V(A) \cup V(B) = V(A') \cup V(B')$ and also that $P_\ell = Q_\ell$ for $\ell \neq i, j$, one can now verify:

$$\pi(P) \cdot X_P(Q) = \frac{2}{k^2 Z} \frac{\prod_{\ell \neq i,j} N_{\mathsf{ST}}(P_\ell)}{N_{\mathsf{ST}}(A \cup B)} N_{\mathsf{ST}}(A)\, N_{\mathsf{ST}}(B)\, N_{\mathsf{ST}}(A')\, N_{\mathsf{ST}}(B') = \pi(Q) \cdot X_Q(P)$$

The condition that $\pi(P) \cdot X_P(Q) = \pi(Q) \cdot X_Q(P)$ is called *detailed balance*, and when it holds for all P, Q, the Markov chain is said to be *reversible*. Satisfying detailed balance with π ensures that it is an equilibrium distribution because if we start with probabilities distributed by π, the probability of being at any state P after one application of X is given by $\sum_Q \pi(Q) \cdot X_Q(P) = \sum_Q \pi(P) \cdot X_P(Q) = \pi(P)$.

An example comparing the cut-edge distribution of samples created with `RevReCom` with 10,000, 1 million, and 100 million steps to the cut-edge distribution of the true underlying distribution π can be see in Figure 3.

[10] To make terminology simpler, we use the term "districts" to refer either to the subsets of vertices or to the subgraphs on those subsets.

[11] We can also define the quantity $\mathsf{sp}(P) := \ln\left(\prod_{i=1}^{k} N_{\mathsf{ST}}(P_i)\right)$ as the *spanning tree score* of a districting plan. This has been proposed in its own right as a measure of the compactness of a plan, with higher scores indicating greater compactness, as in [DT24]. This lets us write π in a format familiar in Markov chain theory as $\pi(P) := \frac{e^{\mathsf{sp}(P)}}{Z}$.

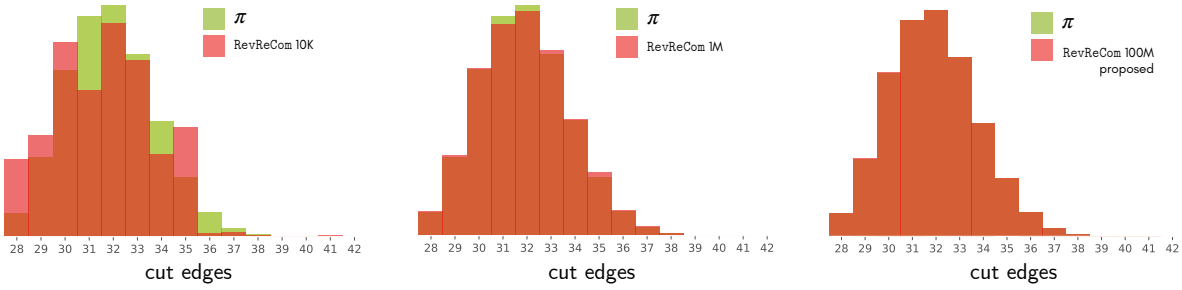


Figure 3: **Convergence in distribution.** For the 7×7 grid divided into 7 equal-sized districts, these plots show the distribution of cut edges in an ensemble of districting plans created after 10,000, 1 million, and 100 million `RevReCom` proposal steps (red) compared to the distribution of cut edges in all districting plans, weighted by π (green). The million-step sample was collected in under ten seconds on a laptop computer, and the time growth is linear.

3.2 Approximate balance

Next, we treat the case where we only require approximate balance for the populations across districts, rather than exact balance as on grid-graph examples. Real-world examples call for this relaxed approximate-balance formulation.

Given population weights on the vertices $w : V \to \mathbb{R}$ and a small population deviation tolerance $\epsilon > 0$, a graph partition P into k districts is *ϵ-approximately balanced* if for all districts P_i,

$$(1-\epsilon)\frac{w(G)}{k} \leq w(P_i) \leq (1+\epsilon)\frac{w(G)}{k},$$

where $w(G) = \sum_{v \in V(G)} w(v)$.

In this setting, a tree may have multiple edges whose removal separates the vertices into two approximately balanced parts. We call an edge e of a tree T an *ϵ-balance edge* if its two complementary components have population balance within tolerance ϵ; when $\epsilon = 0$, this is exactly the notion of a balance edge introduced in the main text.

We modify the algorithm as follows. Let m be a global upper bound on the number of ϵ-balance edges that can exist in any tree spanning a double-district-sized subgraph of G. We will accept a candidate balance edge with probability $1/m$. If there are b balance edges in a particular spanning tree T, this means there is a probability $1 - \frac{b}{m}$ that no edge will be chosen, and the proposal will be rejected at this stage. Recalculating the transition probability:

$$X_P(Q) = \frac{2}{k^2}\left(\frac{N_{\mathsf{ST}}(A', B')}{N_{\mathsf{ST}}(A \cup B)}\right) \cdot \frac{1}{m \cdot E(A', B')} = \frac{2}{mk^2} \cdot \frac{N_{\mathsf{ST}}(A')\, N_{\mathsf{ST}}(B')}{N_{\mathsf{ST}}(A \cup B)},$$

and detailed balance follows as before. Because of the rejection probability of $1 - \frac{b}{m}$, it is advantageous to find an upper bound m that is close to tight. Certainly one can set m to the total number of edges, but this unnecessarily shrinks the acceptance probability. In practice, we have chosen a large value of m, tolerating a high rejection rate in order to be more confident that our bound is globally valid. In our Pennsylvania and Virginia runs, for example, the mean value of m/b (where b is the actual observed number of balance edges in a given proposal) is between four and five.

If the chosen m is not a global upper bound, the process may not converge to π. Even if the number of balance edges is never observed to exceed m in a given run, the resulting limiting distribution is

conditioned on never encountering a spanning tree that has more than m balance edges. This could differ from π in subtle ways, an issue that equally impacts `RevReCom` and the SMC sampler.

To guarantee the existence of unique steady state that attracts every starting distribution for a Markov chain, we must verify that the chain is ergodic, i.e., aperiodic and irreducible. In general, irreducibility for recombination chains is a hard open problem, though there is a great deal of recent work in that direction. See §B.1 for an extended discussion of ergodicity.

4 Empirical Results

4.1 Benchmarking with summary statistics

Quantities of interest assessed for a plan can be thought of as a projection from the state space to $\mathbb{R}^n$, or just to the real numbers in the case of a single metric. These projections are of interest both because summary statistics may capture many of the relevant features of a plan (like its partisan performance, its number of county splits, and so on), and because lower-dimensional representations can be easier to visualize and to understand.

In the demonstrations here we use the push-forward to plan-wide numerical scores to present empirical results. Some scores are shape-based, like the cut edge count (Figures 3 above and 4 below). Some are partisan, like the count of districts favoring some political party (Figure 1). In addition to plan-wide scores, we will also study district-level numerical scores like the partisan vote share by district (Figure 5).

It is easy to see that convergence in the push-forward distribution to a summary statistic may occur well before representative sampling in the state space overall. This means that the quality of convergence in a projected statistic only gives an inequality bound on convergence in the full state space—failure to converge in projection ensures failure to converge overall, but success provides no guarantees. Nevertheless, using a statistic that varies in a way that is related to the stationary distribution will clearly give a more discerning view of convergence than one that is likely to have less or no relationship to π. For this reason, when sampling from the spanning tree distribution, the cut edge plots will be far more likely to catch convergence problems than the partisan plots.

To assess the similarity of sampling distributions, we will use *Wasserstein distance* (also known as earth-mover distance), which may be familiar to readers from optimal transport theory. The Wasserstein distance between two distributions over a common metric measure space computes the cost of shifting the probability mass from one distribution to the other.[12] The Wasserstein distance between two distributions is given by the lowest-cost transport plan; for single quantities, this is the integral of the absolute difference between the two cumulative distribution functions (CDFs), meaning that there is no need to identify a minimum-cost matching to compute the distance.

In Markov chain theory, mixing time is customarily measured with total variation distance, which is a normalization of L^1 distance between probability distributions on the state space. Convergence in total variation distance implies convergence in (L^1) Wasserstein distance, and Wasserstein distance on projected statistics is used here.

4.2 Benchmarking against a complete enumeration

In most practical applications, it is not feasible to give precise bounds on mixing time; instead, practitioners rely on established convergence diagnostics to give heuristic evidence that results are reliable. These include *multi-start tests*, where chains are run from different starting configurations to be sure they are escaping the neighborhood where they were initialized. We also perform *enlargement*

[12] For instance, if a histogram with total mass one is moved by adding six to every recorded value, then the total cost of the move is 6—here, implicitly, neighboring bins are regarded as one unit apart. If some transport plan moves 15% of the mass of a distribution to a location two units away from its start, then its total cost is 0.3.

tests, collecting samples long past the point that statistics appear to stabilize in order to create more opportunity to identify bottlenecks causing pseudo-convergence.

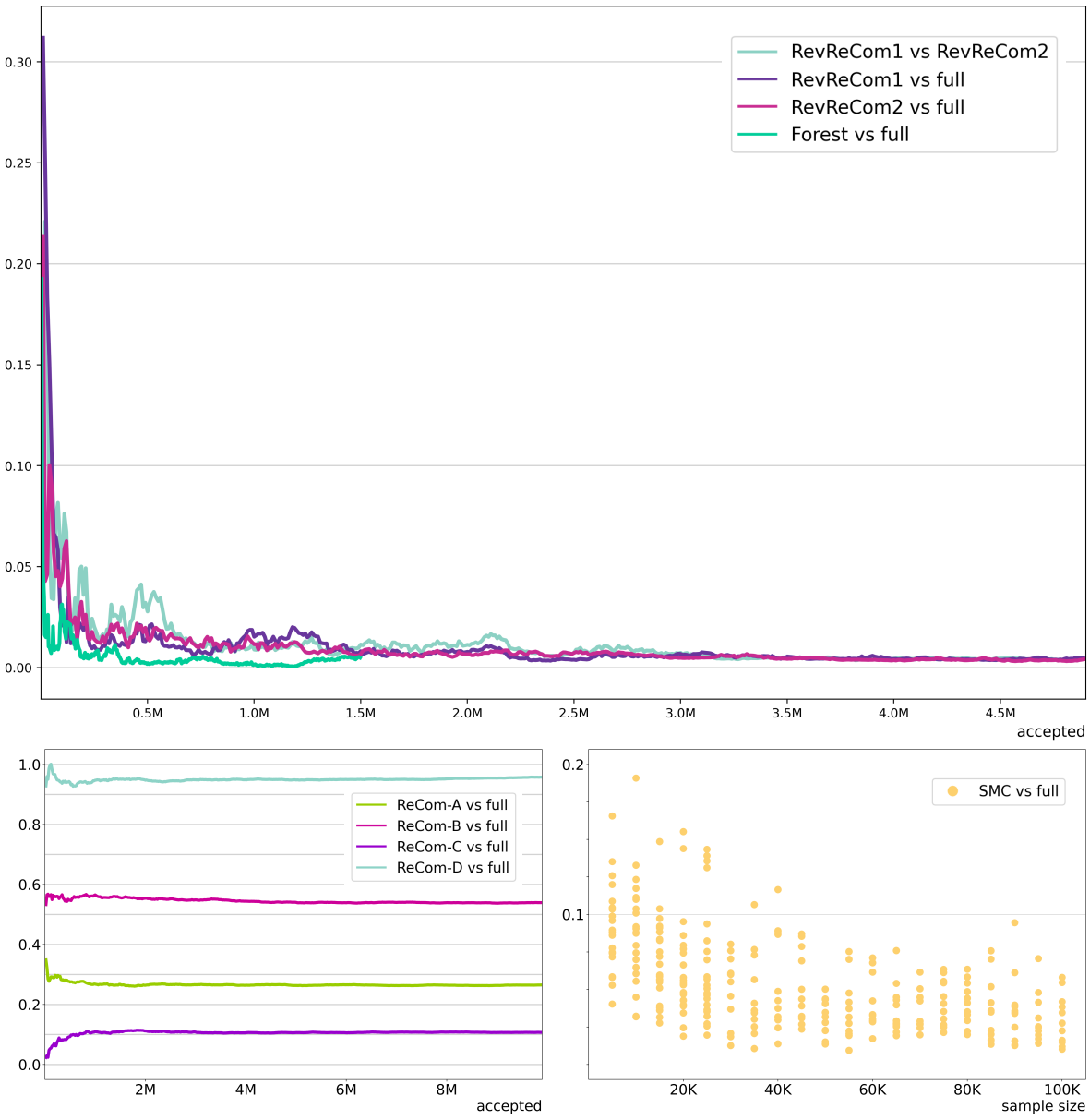


Figure 4: **Convergence diagnostics for the 7×7 grid.** In this figure, we compare different methods at their largest practical sample sizes. TOP: A trace plot of Wasserstein distance between cut edge distributions. Long runs are compared against each other, and against the π-weighted ground truth, showing excellent accuracy for the Markov chain methods. For `RevReCom`, the time to 3 million accepted steps on a laptop is about ten minutes. BOTTOM LEFT: The `ReCom`–A,B,C,D variants stabilize quickly, though they do not exactly converge to π. BOTTOM RIGHT: SMC is not a Markov chain, so outputs are shown here with dots representing whole runs. Performance improves with batch size. See also Figure 9.

We will use the $7 \times 7 \to 7$ districting problem as a test case: we partition a 7×7 grid into 7

districts of 7 units each, which gives precisely 158, 753, 814 configurations. To date, this is the largest districting problem that is readily manipulable in its entirety.[13] The simplest summary statistic that is closely related to the target distribution is the count of cut edges in the partition, which can vary from a minimum of 28 to a maximum of 42. Figure 3 compares the distribution of cut edges (the sizes of cut-sets) in the growing ensemble to the distribution under π. The 100-million-step sample is visually indistinguishable from the ground-truth histogram.

Figure 4 (top) shows all three pairwise Wasserstein distances between the ground-truth spanning tree distribution on the $7 \times 7 \to 7$ problem and two runs of `RevReCom` from different seeds, using the cut edge count as the summary statistic. The bottom left plot shows Wasserstein distance to π for runs of `ReCom`-A,B,C,D. The SMC plots (bottom right) use scatterplots rather than a traceplot, because SMC ensembles are built in batches rather than stepping from plan to plan over time. But the idea is similar: each point shows the Wasserstein distance from the cut edge distribution of an SMC ensemble to the ground-truth π-weighted distribution.

A key takeaway from Figure 4 is that the time it takes the two ensembles generated by `RevReCom` to be similar to each other tracks with the time it takes for each to be similar to the full enumeration. This is encouraging for the value of the multi-start heuristic when there is no full enumeration available.

4.3 Benchmarking at full scale

In districting at the state level, the number of possible districting plans is so large that there is no hope of enumerating them all to get a ground-truth distribution. However, we can still compare runs across different seeds, as in Figure 5.

Because our goal is to compare k-piece partitions, it is also natural to use statistics that attach a numerical value to each district rather than to the plan as a whole. To illustrate, we generate a plot in Figure 5 based on the Clinton share of the major-party presidential vote in 2016 across the districts of Virginia Congressional plans. For a given plan, we then sort these values from lowest to highest and treat the statistic as vector-valued. For instance, one districting plan might give the ordered Clinton shares $(.29, .37, .41, .46, .49, .52, .55, .59, .62, .66, .75)$ across the eleven districts. In this way we can compare plans by comparing their lowest Clinton-share district, indexed 1 across the ensemble, or the second-lowest, and so on.[14] We can also visualize this in a boxplot, shown in Figure 5 (bottom) for the Virginia Congressional plans; the boxes show the 25th-75th percentile range for the value in that indexed district, the whiskers capture the 1st-99th percentile range, and the median is marked.

To get an overall convergence diagnostic, we can apply L^1 Wasserstein distance to these district-level values. An ensemble of plans with a district-level numerical score defines k distributions over the real numbers: the first distribution is given by the values at index 1 (which are the smallest in their plans, by our order convention), the next at index 2, and so on. To compare two vector-valued distributions, one first computes the k coordinatewise Wasserstein distances. Summing these gives the L^1-Wasserstein distance. This is a measure of the distance between two distributions, and can be computed whether the distributions come from a sample, as in an ensemble, or from a complete enumeration. A traceplot can then be used to show the distances between time-t ensembles and a static distribution, or between two time-t ensembles, as in Figure 5 (top).

For Figure 5, the starting plan is the Congressional districting plan from either 2012 or 2016, and we run the latter with two different random-number seeds. The runtime is under five hours. The similarity in Democratic vote statistics across the three runs lends support to the idea that 1 million accepted proposals is sufficient to produce high-quality partisan statistics by district (see also Figure 9, which suggests that even shape statistics, which are more discerning, can also be accurately obtained in tractable time).

[13] Imai et al. have rightly emphasized the importance of using large examples for validation [FIKK20, MI23], but they have tended to use much smaller test cases. In addition, earlier demonstrations often use summary statistics unrelated to the target distribution, which have lower diagnostic value for convergence.

[14] It is important to remember that this means we are comparing districts that may be geographically unrelated.

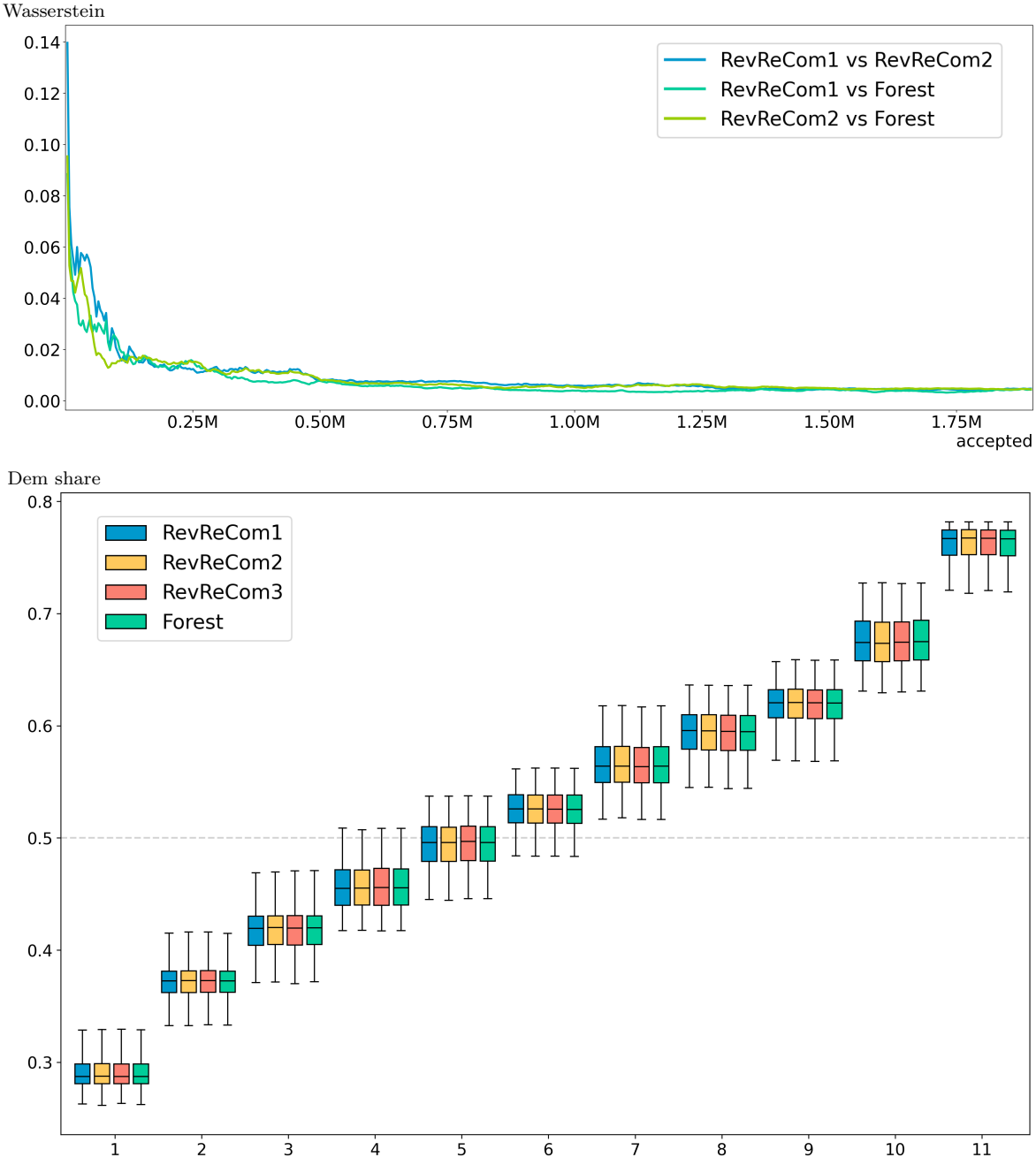


Figure 5: **Virginia example.** TOP: The L^1 Wasserstein trace plot for Democratic vote share by district (from the 2016 Presidential contest) across three different pairwise comparisons of `RevReCom` and Forest `ReCom` runs. BOTTOM: The corresponding box-and-whiskers plot showing the partisan shares by district. The left-most column shows the range of shares in the least Democratic district in each plan; the next column in the second-least Democratic; and so on. In this race, Clinton received 52.8% of the major-party share, and this plot shows that in a π-typical plan, 6/11 districts would have a Democratic advantage, 4/11 would have a Republican advantage, and the last would be very competitive, if people voted as they did in this (Clinton vs. Trump) vote pattern.

4.4 Heatmaps and cross-validation

The modifications that let `RevReCom` target π are calibrated rejection steps; not only is the acceptance rate low, but convergence also requires many accepted steps.[15] Thus the asymptotic guarantees come at a significant cost—not only clock time, but also handling much larger datasets—which makes this `RevReCom` chain unlikely to overtake the faster `ReCom` chains, or SMC runs with smaller ensemble sizes, in practical use. However, it still serves several crucial purposes: very long runs of `RevReCom` give the clearest picture yet of the true spanning tree distribution, and this helps us to better evaluate the sampling distributions produced by other methods.

Comparing different methods on the same redistricting problems allows for comparisons that go beyond convergence time. One visualization that captures some significant differences is a heatmap that shows how often nodes are reassigned to new districts in the course of a Markov chain run. In Figure 6, two grid graphs are constructed with widely varying node weight to exaggerate the disparities in population across nodes that can be observed in realistic districting problems. These graphs model the partition problem on two multiscale regions that divide homogeneous rectangles into squares of different sizes. For the first example, each quadrant of a square has equal population, but the quadrants are subdivided into a 2×2, 4×4, 8×8, and 16×16 grids, respectively. That means that the dual graph of this tiling has 340 vertices, with the largest nodes having 64 times the population of the smallest. This graph will be partitioned into 4 districts of equal size. The second multiscale grid graph in this experiment has 5460 nodes, with six sections divided $2 \times 2, 4 \times 4, \ldots, 64 \times 64$, giving the largest nodes 1024 times the population of the smallest. This graph will be partitioned into 6 districts of equal size. (Disparities this large can be found in real geography. In Iowa, the largest county has roughly 140 times the population of the smallest; in Texas, the ratio is over 100,000.) We run millions of steps of each of the six Markov chains and record how often each node is reassigned.

We find that some pairs of chains (`ReCom`-A and `ReCom`-C, or `ReCom`-B and `ReCom`-D) have visually indistinguishable patterns of reassigning nodes, though those four ultimately arrive at clearly different stationary distributions. On the other hand, Forest `ReCom` and `RevReCom` have quite different reassignment patterns, while they converge to precisely the same steady state. This should raise our confidence in the statistics produced by Forest `ReCom` and `RevReCom`, when they agree—they are truly distinct samplers, and so their findings are not redundant but are mutually cross-validating.[16]

4.5 Convergence considerations

`RevReCom`, Forest `ReCom`, and SMC all have (caveated) asymptotic guarantees that their sampling distributions converge to π given large enough samples. We briefly review the caveats that come with those guarantees and the feasibility of closely approximating π-distributed sampling in practice. `RevReCom` and SMC both use an upper bound m on the number of balance edges, described in Section 3.2. The Markov chain methods face questions of whether their state space is irreducible (i.e., connected), which is required for uniqueness of the stationary distribution. SMC faces quickly exploding size requirements for sampling as the number of districts gets large. All of these produce issues that might bias the distribution of a sample collected under resource constraints.

Beyond the spanning tree distribution, all three methods can be set to target any distribution in principle, but may pay a steep price in convergence time for targeting distributions far from π. For instance, Chatterjee and Diaconis show in [CD18] that importance sampling methods like SMC will require an exponential increase in sample size as the target distribution deviates from the generating distribution (in KL divergence). Similarly, `RevReCom` and Forest `ReCom` may experience steep increases in rejection rates when targeting other distributions, requiring longer runs to obtain the same number of accepted proposals. The discussion of convergence obstructions continues in §B.1.

[15] On the 7×7, 16.4% of proposals (about one in six) were accepted in one benchmarking run. In the state-level runs, roughly 0.05% of proposals (or one in 2000) were accepted. See Supplemental Table 2.

[16] This is subject to the caveat above and in §B.1 about ergodicity; if the state space is disconnected and both Markov chain methods are initialized in the same component, then the cross-validation only applies to that component.

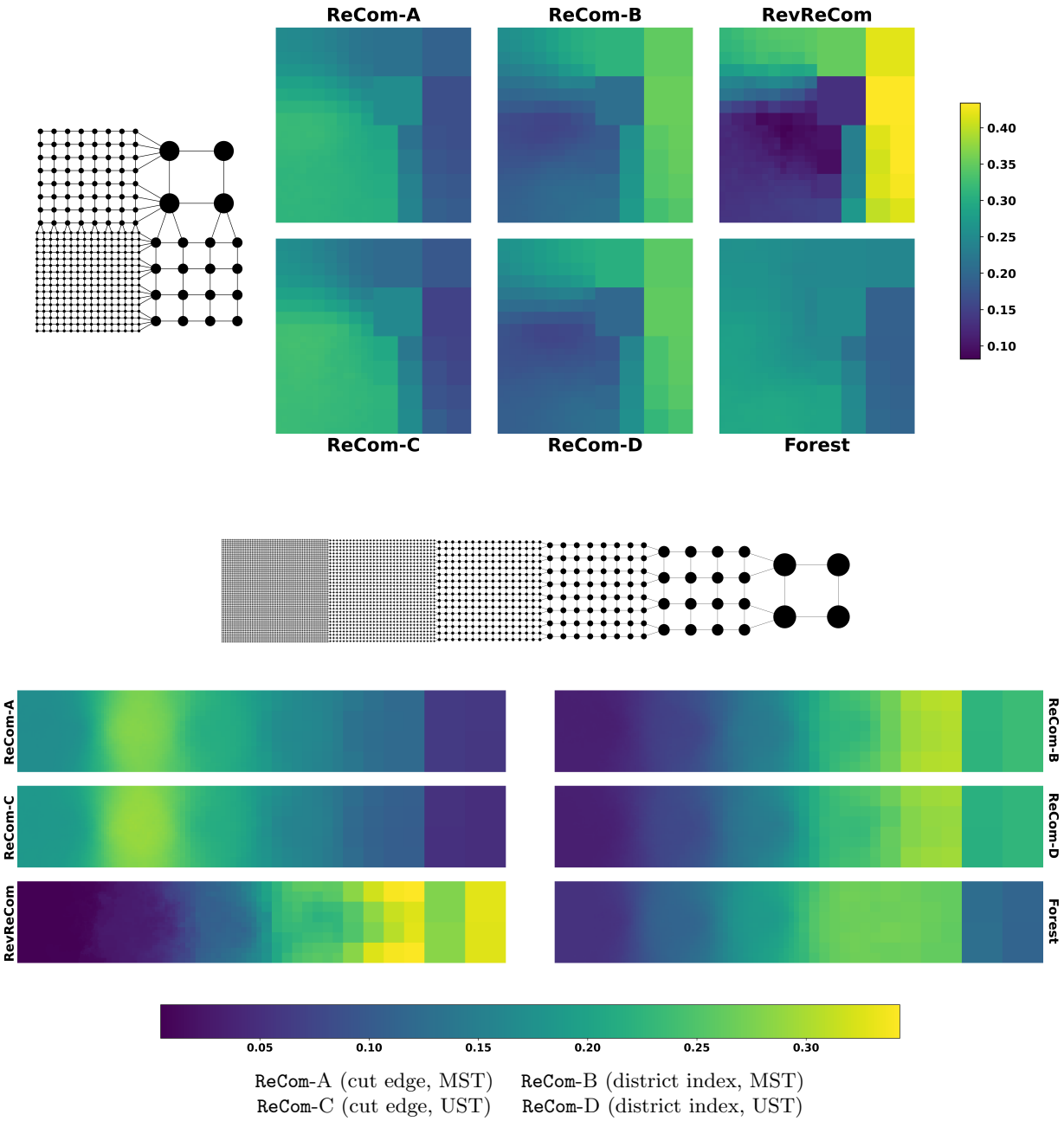


Figure 6: **Multiscale examples.** The heatmaps provide a visualization of how each random walk transits the state space of balanced partitions by showing how often each node is reassigned in the first 50,000 accepted proposals. We see that `ReCom`-A and `ReCom`-C, which both use cut edges to select districts to fuse, have very similar behavior, whether UST or MST is used to choose spanning trees; similarly for `ReCom`-B and `ReCom`-D, which both use random indices to choose districts. `RevReCom` and Forest `ReCom` flip different vertices but arrive at precisely the same steady state.

5 Discussion

The method of ensembles allows us to measure the consequences as the rules of redistricting interact with the political geography of votes, supplying policymakers and courts with a much-needed means to study the central tendencies of districting.[17] But they do not commit us to a view that *neutral is fair*. For instance, it is widely believed that some kind of urban/rural effect tends to create significant structural partisan advantages for a party with a more rural base, as we saw with Republicans in Pennsylvania. Generally, it is increasingly clear that neutrally drawn single-member plurality districts give systematic underrepresentation to racial and ethnic minority groups as they are currently geographically distributed in the United States [DS21]. Reasonable observers can disagree about how to define fair redistricting: is a "blind" plan necessarily fair? Or, to name another alternative, policymakers might prefer plans that tend to favor representation that is proportional to popular preference—and randomized runs can help us understand if that is feasible within a given framework of rules.[18]

Spanning tree methods for sampling are appealing to courts for good reasons, including the growing scientific consensus around their construction and their increasing accessibility with lightweight, open-source computing. Using spanning trees in the re-partition step controls the sizes of cut-sets without introducing additional parameters or weights, minimizing the need for tuning and the room for gaming basic outputs. Formalizing the use of the spanning tree distribution as the canonical choice for weighting partitions lets us mutually compare methods under the fundamental constraints of population balance, contiguity, and a preference for compactness. From there, if the specific setting calls for it, one can layer in weighting terms or other mechanisms that take local rules and priorities into account, having validated that the basic engine is effective enough to deliver on its asymptotic guarantees in practical time. County preservation, city preservation, increasing minority groups' opportunity-to-elect, avoiding incumbent pairings, resembling a prior map, or even respecting communities of interest collected through a public feedback process—all have been operationalized in ways that are compatible with these algorithmic methods.[19] Reversible recombination, created with a small modification to the original recombination chains, is the most powerful tool yet proposed, has formally verifiable properties, and gives particular insight into methods already in wide use in courts and in public policy.

The data demonstrations presented here confirm that the methods under study—the original recombination variants, Forest `ReCom`, SMC, and `RevReCom`—are all capable of producing reliable estimates for key summary statistics at the state level. SMC is at its best with small numbers of districts; with more than about ten districts, it can require forbiddingly large sample sizes to produce π-distributed samples (see Figure 9), and other distributions will be still harder to target. Forest `ReCom` and `RevReCom` have the strongest accuracy performance (which can only be measured rigorously when ground truth is known), and produce mutually supporting estimates on both small and full-scale problems; `RevReCom` scales best, passing convergence checks on the larger problems in reasonable time.

Taken together, ensemble methods give us essential tools for the 21st century as we engage in continued debate about our requirements, and our aspirations, for representative democracy. The development of spanning tree methods to understand redistricting is a model for scientific engagement in policy: purpose-built tools have made clear progress on a pressing real-world problem.

[17] Moreover, because Markov chain methods are easily amenable to heuristic optimization methods (hill-climbing, annealing, short bursts [CGHG$^+$23], and so on), we can also produce runs that seek to drive various scores and summary statistics up or down. Local search methods give a view of the "elasticity" of districting outcomes by probing how effective it is to guide the exploration process. In this way we can start to understand, to the extent that we might have districting goals beyond the simple neutral consequences of the rules, how attainable those goals might be.

[18] See, for instance, [DS22].

[19] That is not to say that these districting criteria have authoritative or one-size-fits-all interpretations, so it will remain important to have a canonical distribution that handles the most fundamental criteria.

References

[ACH+21] Eric A. Autry, Daniel Carter, Gregory Herschlag, Zach Hunter, and Jonathan C. Mattingly. Metropolized multiscale forest recombination for redistricting. *Multiscale Modeling & Simulation*, 19(4):1885–1914, 2021.

[ACH+23] Eric Autry, Daniel Carter, Gregory Herschlag, Zach Hunter, and Jonathan C. Mattingly. Metropolized forest recombination for Monte Carlo sampling of graph partitions. *SIAM Journal on Applied Mathematics*, 83(4):1366–1391, 2023.

[AD86] David Aldous and Persi Diaconis. Shuffling cards and stopping times. *The American Mathematical Monthly*, 93(5):333–348, 1986.

[AF02] David Aldous and James Allen Fill. Reversible Markov chains and random walks on graphs, 2002. Unfinished monograph, recompiled 2014, available at `http://www.stat.berkeley.edu/~aldous/RWG/book.html`.

[AKK+22] Hugo A. Akitaya, Matias Korman, Oliver Korten, Diane L. Souvaine, and Csaba D. Tóth. Reconfiguration of connected graph partitions via recombination. *Theoretical Computer Science*, 923:13–26, 2022.

[Ald87] David Aldous. On the Markov chain simulation method for uniform combinatorial distributions and simulated annealing. *Probability in the Engineering and Informational Sciences*, 1:33–46, 1987.

[BD92] Dave Bayer and Persi Diaconis. Trailing the dovetail shuffle to its lair. *The Annals of Applied Probability*, 2(2):294–313, 1992.

[Bro06] A. E Brockwell. Parallel Markov chain Monte Carlo simulation by pre-fetching. *Journal of Computational and Graphical Statistics*, 15(1):246–261, 2006.

[BS22] Amariah Becker and Justin Solomon. Redistricting algorithms. In Moon Duchin and Olivia Walch, editors, *Political Geometry*, chapter 16, pages 303–340. Birkhäuser Books, 2022.

[Can24] Sarah Cannon. Irreducibility of recombination Markov chains in the triangular lattice. *Discrete Applied Mathematics*, 347:75–130, 2024.

[CBH+21] Jeanne N. Clelland, Nicholas Bossenbroek, Thomas Heckmaster, Adam Nelson, Peter Rock, and Jade VanAusdall. Compactness statistics for spanning tree recombination. Preprint. Available at `https://arxiv.org/abs/2103.02699`, 2021.

[CD18] Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *Annals of Applied Probability*, 28(2):1099–1135, 2018.

[CDD24] Sarah Cannon, Daryl DeFord, and Moon Duchin. Repetition effects in a Sequential Monte Carlo Sampler. Preprint. Available at `https://arxiv.org/abs/2409.19017`, 2024.

[CGHG+23] Sarah Cannon, Ari Goldbloom-Helzner, Varun Gupta, JN Matthews, and Bhushan Suwal. Voting rights, Markov chains, and optimization by short bursts. *Methodology and Computing in Applied Probability*, 25(1), 2023.

[CLLV22] Moses Charikar, Paul Liu, Tianyu Liu, and Thuy-Duong Vuong. On the complexity of sampling redistricting plans. Preprint. Available at `https://arxiv.org/abs/2206.04883`, 2022.

[CPTF24] Sarah Cannon, Wesley Pegden, and Jamie Tucker-Foltz. Sampling balanced forests of grids in polynomial time. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1676–1687, New York, NY, USA, 2024. Association for Computing Machinery.

[DDS21] Daryl DeFord, Moon Duchin, and Justin Solomon. Recombination: A Family of Markov Chains for Redistricting. *Harvard Data Science Review*, 3(1), 2021.

[DFJN14] Charles R. Doss, James M. Flegal, Galin L. Jones, and Ronald C. Neath. Markov chain Monte Carlo estimation of quantiles. *Electronic Journal of Statistics*, 8(2):2448–2478, 2014.

[Dia88] Persi Diaconis. *Group Representations in Probability and Statistics*, volume 11 of *Monograph Series*. Institute of Mathematical Statistics Lecture Notes, 1988.

[Dia09] Persi Diaconis. The Markov chain Monte Carlo revolution. *Bulletin of the American Mathematical Society*, 46:179–205, 2009.

[DS81] Persi Diaconis and Mehrdad Shahshahani. Generating a random permutation with random transpositions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 57(2):159–179, Jun 1981.

[DS21] Moon Duchin and Douglas M. Spencer. Models, Race, and the Law. *The Yale Law Journal*, 130:744–797, 2021.

[DS22] Moon Duchin and Gabe Schoenbach. Redistricting for proportionality. *The Forum*, 20(3-4):371–393, 2022.

[DT24] Moon Duchin and Bridget Eileen Tenner. Discrete geometry for electoral geography. *Political Geography*, 109:103040, 2024.

[Duc22] Moon Duchin. Introduction. In Moon Duchin and Olivia Walch, editors, *Political Geometry*, chapter 0, pages 1–28. Birkhäuser Books, 2022.

[Fel68] William Feller. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley, 1968.

[FIKK20] Benjamin Fifield, Kosuke Imai, Jun Kawahara, and Christopher T. Kenny. The essential role of empirical validation in legislative redistricting simulation. *Statistics and Public Policy*, 7(1):52–68, 2020.

[FP23] Alan Frieze and Wesley Pegden. Subexponential mixing for partition chains on grid-like graphs. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3317–3329, 2023.

[Gey92] Charles J. Geyer. Practical Markov Chain Monte Carlo. *Statistical Science*, 7(4):473–483, 1992.

[GR92] Andrew Gelman and Donald B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992.

[Her] Gregory Herschlag. mergesplitcodebase. Python Library and Julia Library. `https://git.math.duke.edu/gitlab/gjh/mergesplitcodebase` `https://git.math.duke.edu/gitlab/quantifyinggerrymandering/multiscalemapsampler-public`.

[KLM89] Richard M Karp, Michael Luby, and Neal Madras. Monte-Carlo approximation algorithms for enumeration problems. *Journal of Algorithms*, 10(3):429–448, 1989.

[KMFI] Christopher Kenny, Cory McCartan, Ben Fifield, and Kosuke Imai. redist: Simulation methods for legislative redistricting. GitHub Repository. `https://github.com/alarm-redist/redist/`.

[KW08] J. Kim and T. Wilhelm. What is a complex graph? *Physics A*, 387:2637–2652, 2008.

[Lyo05] Russell Lyons. Asymptotic enumeration of spanning trees. *Combinatorics, Probability, and Computing*, 14(4):491–522, 2005.

[MGG] MGGG Redistricting Lab. Gerrychain. Python Library. `https://github.com/mggg/GerryChain`.

[MI23] Cory McCartan and Kosuke Imai. Sequential Monte Carlo for sampling balanced and compact redistricting plans. *The Annals of Applied Statistics*, 17(4):3300 – 3323, 2023.

[MKS+22] Cory McCartan, Christopher T. Kenny, Tyler Simko, George Garcia, Kevin Wang, Melissa Wu, Shiro Kuriwaki, and Kosuke Imai. Simulated redistricting plans for the analysis and evaluation of redistricting in the united states. *Scientific Data*, 9(1):689, 2022.

[Moo17] Cristopher Moore. The computer science and physics of community detection: Landscapes, phase transitions, and hardness. *Bull. EATCS*, 121, 2017.

[NDS] Elle Najt, Daryl DeFord, and Justin Solomon. Complexity and geometry of sampling connected graph partitions. Available at `https://arxiv.org/abs/1908.08881`.

[NDS21] Elle Najt, Daryl DeFord, and Justin Solomon. Empirical sampling of connected graph partitions for redistricting. *Phys. Rev. E*, 104:064130, Dec 2021.

[Nel22] Garrett Dash Nelson. The elusive geography of communities. In Moon Duchin and Olivia Walch, editors, *Political Geometry*, chapter 11, pages 221–234. Birkhäuser Books, 2022.

[POM09] Mason A. Porter, Jukka-Pekka Onnela, and Peter J. Mucha. Communities in networks. *Notices of the American Mathematical Society*, 56(9):1082–1097, 1164–1166, 2009.

[PTF22] Ariel D. Procaccia and Jamie Tucker-Foltz. Compact redistricting plans have many spanning trees. *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2022.

[RR] Peter Rock and Parker Rule. Reversible recom replication. GitHub Repository. `https://github.com/mggg/RRC-Replication`.

[Rul] Parker Rule. Frcw. Rust Implementation of Reversible ReCom. `https://github.com/mggg/frcw.rs`.

[RW22] Jonathan Rodden and Thomas Weighill. Political geography and representation: A case study of districting in pennsylvania. In Moon Duchin and Olivia Walch, editors, *Political Geometry*, chapter 5, pages 101–127. Birkhäuser Books, 2022.

[Tap24] Kris Tapp. Spanning tree bounds for grid graphs. *Electronic Journal of Combinatorics*, 31(1), 2024.

[TF23] Jamie Tucker-Foltz. Locked polyomino tilings. Preprint. Available at `https://arxiv.org/abs/2307.15996`, 2023.

[VGS+21] Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved $\hat{R}$ for assessing convergence of MCMC (with discussion). *Bayesian Analysis*, 16(2), June 2021.

[Wil96] David Bruce Wilson. Generating random spanning trees more quickly than the cover time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC, page 296–303, 1996.

6 Acknowledgements

The authors are grateful to the developer team behind GerryChain, with special thanks to Daryl DeFord, Max Hully, JN Matthews, Bhushan Suwal, Anthony Pizzimenti, Max Fan, and Peter Rock, as well as other members and collaborators of the Data and Democracy Lab. We thank Kosuke Imai for suggesting that we include sequential Monte Carlo (SMC) comparisons alongside Markov chain methods, and also thank Cory McCartan for his help making sure we were using SMC as intended.

Funding. The authors are deeply grateful for funding support from multiple sources. S.C. acknowledges NSF grants DMS-1803325 and CCF-2104795; M.D. acknowledges NSF DMS-2005512; D.R. acknowledges NSF grants CCF-1733812 and CCF-2106687. S.C., M.D., and D.R. are grateful to the Simons-Laufer Mathematical Sciences Institute, where we were research members together while this manuscript was in preparation.

Author Contributions. All authors were involved in conceptualization, methodology, investigation, and writing. M.D. and P.R. were engaged in data curation, and P.R. is the principal software developer for this project. Author order is alphabetical, following the convention in mathematics.

Competing Interests. The authors have no competing interests. M.D. served as an expert in *League of Women Voters v. Pennsylvania* (2018), the case that provides context for Figure 1, as well as in *Allen v. Milligan* (2023) and numerous other cases in the current redistricting cycle.

Data and Materials Availability.

Code for these experiments is publicly available at `https://github.com/mggg/RRC-Replication` and `github.com/mggg/frcw.rs`.

A Legal reception

By looking at just two recent cases—*Rucho v. Common Cause*, a 2019 partisan gerrymandering case from North Carolina, and *Allen v. Milligan*, a 2023 racial vote dilution case from Alabama—we can find all nine current U.S. Supreme Court justices giving weight to redistricting algorithms as helpful evidence, and specifically citing work that uses Markov chain methods. The selections quoted here show an emphasis on finding examples or a benchmark, not a putatively best or optimized choice. In addition, courts have looked for reassurance that this baseline for comparison does not depend sensitively on idiosyncrasies of the method.

Kagan in *Rucho* dissent (joined by Sotomayor, Ginsburg, and Breyer)

- "The approach... begins by using advanced computing technology to randomly generate a large collection of districting plans that incorporate the State's physical and political geography and meet its declared districting criteria, except for partisan gain. ... The further out on the tail, the more extreme the partisan distortion and the more significant the vote dilution." (p18-19)
- "The point is that the assemblage of maps, reflecting the characteristics and judgments of the State itself, creates a neutral baseline from which to assess whether partisanship has run amok." (p23-24)

Roberts in *Milligan* majority opinion (joined by Kagan, Sotomayor, Jackson, and Kavanaugh)

- Evidence toward the Gingles 1 precondition: "the 'randomized algorithms' [plaintiffs' expert] employed 'found plans with two majority-black districts in literally thousands of different ways.'" (footnote 7, p26)
- Kavanaugh's concurring opinion asserts that "computer simulations might help detect the presence or absence of intentional discrimination" (p3).
- The Roberts opinion is open to the use of ensembles in voting rights cases (p28), while opting not to *require* them for claims under § 2 of the Voting Rights Act, which is concerned with minority groups' opportunity to elect candidates of choice.

Thomas in *Milligan* dissent (joined by Barrett, Gorsuch, and Alito)

- "In arguing that a vote-dilution claim requires judging a State's plan relative to an undiluted benchmark to be drawn from the totality of circumstances—including, where probative, the results of districting simulations—the State argues little more than what we have long acknowledged." (footnote 15, p31)
- Thomas calls `ReCom`-based ensemble evidence from [DS21] "surely probative" (p23).
- Alito also writes in a separate opinion that ensembles provide "strong evidence" for the Alabama case (p13).
- In fact, the dissenting bloc writes in favor of *requiring* this kind of ensemble evidence for a Voting Rights Act claim.

B Theory

B.1 Potential obstructions to convergence

B.1.1 Reducibility

From the detailed balance expressions given in §3.1 for the perfect balance case and in §3.2 for the approximate balance case, we know that the spanning tree distribution π is a steady state of `RevReCom`. However, the fundamental theorem that guarantees that this is the unique stationary distribution, and that all other distributions converge to π under iterations of the Markov process, requires the hypothesis that the system is *ergodic*, i.e., aperiodic (the gcd of all lengths of closed cycles is 1) and irreducible (it is possible to transition from any state to any other state in finite forward time). In all Markov chains used to sample graph partitions on real-world data, including all the ones described here, aperiodicity is immediate because there are large cliques, but proofs of irreducibility have remained elusive. We note that `ReCom` has the same state space as `RevReCom`, with the same allowed transitions but different transition probabilities, so irreducibilty for `ReCom` and `RevReCom` are equivalent. Recombination moves include flips as a special case, so the state spaces for flip chains are strictly more likely to be reducible, i.e., disconnected.

There is mounting evidence that recombination chains are irreducible under realistic sampling conditions on real-world dual graphs, despite the difficulties in proving it. The main challenge is the population constraints: the tighter the allowed deviation, the more likely it is that the valid moves no longer connect the state space. As population constraints loosen, irreducibility is known: Akitaya et al. have shown that the chain is irreducible when the dual graph is Hamiltonian and districts can shrink arbitrarily small and grow up to double their ideal size [AKK$^+$22]. This is because allowed moves can create some large districts and several single-vertex districts, and moving through such configurations allows greater flexibility. A recent result of Cannon proves irreducibility under much tighter balance conditions (± 1 from ideal size), but is limited to the special case of $k = 3$ districts on a triangular lattice [Can24]. There are also positive irreducibility results by Charikar et al. under tight size constraints for specific, highly structured examples [CLLV22]. Many of the negative results are for unrealistic, stylized graph "gadgets," such as by Akitaya et al. for planar graphs with many more large faces than one would expect in a typical geography dual graph [AKK$^+$22].

Small rectangular grids produce many interesting examples when there are few units per district. Tucker-Foltz finds a range of interesting behavior in small finite grid examples where districts have 3-5 units each [TF23]. He gives several examples of tilings of finite regions by m-ominoes, for $m = 3, 4, 5$, that are *recombination rigid*, meaning that they are not connected by any perfect-balance recombination transition to any other configuration, so they are isolated points in the state space; he also shows that certain grid examples, like the $6 \times 6 \to 12$ problem, have state spaces with multiple large connected components. The $6 \times 6 \to 3$ problem (a 6×6 grid divided into $k = 3$ exactly balanced districts of 12 units each) is known to be ergodic for recombination, but every known proof involves extensive case analysis. For the $6 \times 6 \to 6$ and $7 \times 7 \to 7$ problems, all known proofs of irreducibility use brute-force computation. This direct verification of irreducibility is computationally infeasible for the larger dual graphs arising from real-world instances.

Realistic geography dual graphs are usually planar (though exceptions can arise when the pieces themselves are disconnected) and tend to look like near-triangulations of simply connected domains, with hundreds or thousands of units per district. No examples of disconnected state spaces are known when the dual graph has small perimeter compared to interior (as in grids and lattices and all known real-world examples) and the number of units per district is at least 6. Thus, recombination is widely believed to be irreducible on geography dual graphs under legally reasonable values of the population deviation tolerance ϵ, such as $\epsilon = .01$ for Congressional districts and $\epsilon = .05$ for smaller legislative districts.

Without a proof of irreducibility, we only know that a `RevReCom` chain initialized at a particular state converges to the spanning tree distribution on its connected component of the state space. This

provides another reason for verifying that samples seeded at multiple different starting points have similar properties—also known as the multistart heuristic, as in Figure 5. We note that the problem of finding starting points, or seeds, can be a significant challenge facing MCMC practitioners; one possible application of SMC, even long before its samples approximate any target distribution, is to generate starting points for Markov chain exploration.

B.1.2 Ancestor extinction in SMC

Due to its iterative structure of marking districts, the SMC method from [MI23] faces certain convergence obstructions that are not faced by the Markov chain methods. Each of the k districts is selected in a separate "generation" of draws from partial partitions, creating the very likely prospect of so-called *ancestor extinction*: multiple plans in the ith generation will likely be drawn from the same parent in generation $i-1$, and the concentration compounds generationally, so that a set of identical districts are likely to be found in common in a high share of plans finally produced, while other districts drawn in early generations are never found in the final output. For example, a recent expert report in New York legislative redistricting used SMC to generate samples of state Senate plans with $k = 63$ districts. A rebuttal report replicated the SMC ensemble used by the plaintiffs' expert and found that 31 districts out of 63 were repeated *exactly identically* in most of the plans (3129 out of 5000) (Affidavit of Kristopher Tapp in *Harkenrider v. Hochul*, see brief).

The SMC authors are aware of these issues and take mitigating steps in their own work. One such such step is to *modularize* large states into smaller pieces in an arbitrary way, though this may change the sampling distribution overall. Another step is to combine subsamples from separate batches rather than reporting single batches. Both of these strategies are employed in the Texas, Florida, and California ensembles from their 50-state project, now published in *Scientific Data* [MKS+22]. These techniques are decoupled from theoretical guarantees about the sampling distribution, so we have opted to conduct the largest SMC runs possible (see "largest practical sample sizes," §2.2) and report the resulting ensembles in full.

In addition, SMC uses a correction to account for the fact that it samples labeled rather than unlabeled partitions; this essentially requires a second round of importance sampling to estimate a corrective factor for the first, controlled by an additional parameter called `est_label_mult`. There is an "exploration" parameter `seq_alpha` and various additional parameters to loosen constraints if the process is getting stuck (`pop_temper`, `final_infl`). All of these can help produce diverse-looking samples in a reasonable time, but will likely significantly expand the sample size needed to get batches that resemble draws from π. As a result, we would expect to need batch sizes of many millions in order to get approximately π-distributed samples on state-sized problems, especially with large numbers of districts—currently out of reach of laptop computing. These issues are explored further in [CDD24].

B.2 Constructing Markov chain ensembles

MCMC is frequently used to collect a sample of configurations; as noted in the main text, ensembles collected from MCMC runs will be used to study proposed plans comparatively. Recall from §1.3 that our ensembles will be built by collecting each plan visited by the random walk. We now provide additional details and context for that choice.

Continuous observation. In many applications, researchers will employ parameters s_1 and s_2 to implement *burn-in* and *sub-sampling*: a Markov chain process will skip the first s_1 states before adding a state to the ensemble; subsequently, every s_2^{th} state visited by the chain will be added. In principle, when s_1 is set to the mixing time of the Markov process and s_2 is set as its relaxation time, this produces (approximately) uncorrelated samples from the stationary distribution. (Here the *relaxation time* t_{rel} is the inverse of the spectral gap of the transition matrix; it is closely related to the mixing time t_{mix} at which the sample approaches its steady state.) For real applications that lack rigorous bounds on mixing time, some authors have argued in favor of *continuous observation* that records

every step encountered by the chain. For example, Aldous proves in [Ald87] that for estimating means, continuous observation for $2kt_{rel}$ steps after a burn-in phase is at least as good as taking k samples, each at time $s_2 = 2t_{rel}$ apart. Similarly, Geyer argues in [Gey92] that in practice, one long enough run of the Markov chain with continuous observation suffices for estimating quantities of interest and is conceptually cleaner. Continuous observation is the method we employ, partly because we also consider it helpful in assessing dependence on starting point, which can be hidden by burn-in.

Multiplicity. As is necessary to ensure the desired stationary distribution, we build our ensemble with multiplicity: if we are at state P when a proposal fails, then we count this as a step of the Markov chain and increase the multiplicity of P in our ensemble and in the weighting of the statistics that we gather.

Coverage. The power of Markov chain methods is that they are capable of producing very accurate estimates while visiting only a tiny portion of a state space. However, it is also interesting to consider how many distinct plans are encountered; see Figure 7, which considers the $7 \times 7 \to 7$ state space and shows after ten billion steps only about 18% of states have been visited. Comparing to Figure 3, where we achieved excellent convergence after 100 times fewer steps—and less than 1% coverage—reminds us that the power of MCMC is to approximate a sampling distribution long before cover time.

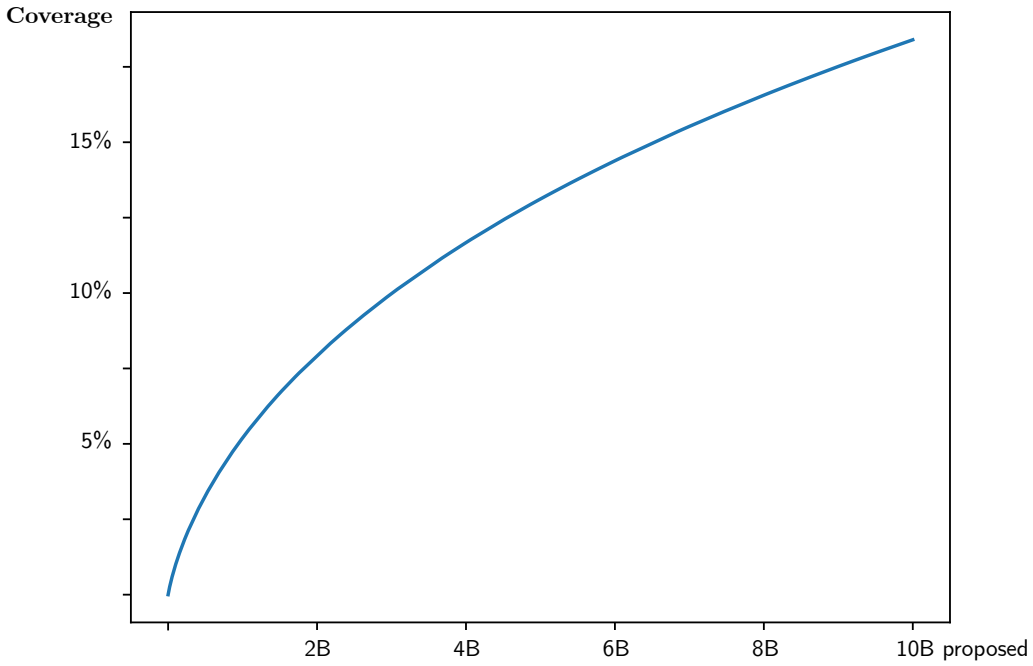


Figure 7: **Coverage of the $7 \times 7 \to 7$ state space.** In ten billion proposed steps on the $7 \times 7 \to 7$ state space, this run of `RevReCom` has visited roughly 18% of states, or about 27 million out of the 150 million distinct plans. It is interesting to compare to Figure 3, where the distribution in a summary statistic looks nearly perfect after only 100M proposal steps.

Sample size. Separately from issues of whether to continuously observe a chain or subsample, there is a large body of literature focusing on how large of a sample is needed to reliably estimate statistics [Fel68, KLM89, DFJN14]. When samples are independent, if the underlying distribution

has mean μ and variance σ^2, the average of n samples has variance σ^2/n. For statistics that take on values in a bounded range, upper bounds on σ^2 are easy to find; for however small one would like the variance of the sample average to be, it is then straightforward to determine the needed sample size. For a random variable X with expectation μ and variance σ^2, by the central limit theorem, the average of n independent random samples of X, which we will denote X_n, is asymptotically normal: $X_n \longrightarrow \mathcal{N}(\mu, \sigma^2/n)$. To ensure that the mean value of some statistic is estimated correctly to one decimal place, one could calculate what is needed for 99% of the probability mass for X_n fall within a range of of size 0.1. In a normal distribution, 99% of the probability mass is between $\mu - 3SD$ and $\mu + 3SD$, where SD is the standard deviation; ensuring $6SD < 0.1$ would give the desired property. For X_n, the standard deviation is $\sigma/\sqrt{n}$, and we would want $6\sigma/\sqrt{n} < 0.1$, or $n > 3600\sigma^2$. Estimates of σ^2, for the original random variable of interest X, are required to know how large n should be. For example, if X is the number of Democratic seats in PA in the underlying data from Figure 1, the empirical variance of X is $\sigma^2 = 0.525$ under the Pres16 votes; using Sen16 votes it is $\sigma^2 = 0.551$. Even rounding up to a variance of 1, this suggests that 3600 truly independent random samples should suffice to ensure, with probability 0.99, that the estimate of the mean of X falls within a range of size 0.1. If instead we wanted to target a range of size 0.01, we would need $6\sigma/\sqrt{n} < 0.01$, or $n > 360,000\sigma^2$. However, this is just for a single statistic, like the mean. For estimating a statistic other than a mean, one can simply define an auxiliary random variable whose mean is the statistic of interest and apply the same analysis to that random variable. If we want to estimate the tails of the distribution, or if we want to make simultaneous estimates for multiple values (as for a district-valued statistic), the needed sample size will go up accordingly, even with perfect access to draws from the desired distribution. Compounding this problem is that samples are correlated in practice. When samples are correlated, the average of n random samples may have much higher variance, so the calculations above should be regarded as a lower bound.

C Implementation

C.1 Implementation of reversible recombination, with parallelization

The implementation of `RevReCom` used in this paper for the empirical results is written in Rust [Rul]. Rust is a programming language increasingly popular in the scientific computing community for its performance and compile-time memory safety guarantees. The implementation uses the general (approximate balance) formulation of `RevReCom` given in §3.2, so that the perfect-balance version of the chain given in §3.1 is a special case for $\epsilon = 0$.

Architecturally, `RevReCom` resembles previous implementations of recombination Markov chains [MGG], but it is optimized to control memory allocation during long runs. Due to these enhancements, the Rust implementation outperforms the principal Python library GerryChain by several orders of magnitude (see Table 3). Furthermore, the memory safety guarantees in Rust make it well suited to multi-threading. We take advantage of this feature to mitigate the high rejection rate of `RevReCom` through a partially parallelized *batching* strategy, which improved the acceptance rate in the benchmarking run to over 300 plans per second.

The choice of spanning tree is made with Wilson's algorithm, which samples uniformly from possible trees (UST) [Wil96]. For an upper bound m on the number of balance edges, we use $m = 30$ in Virginia and Pennsylvania. These values are both far larger than the largest number of balance edges we ever observed during an execution of `RevReCom` on these states.[20] For the PA and VA runs, the batching was executed with 1024 proposals per batch, split across 8 cores.

[20] Recall that the resulting distribution in both these cases could differ slightly from π because it is conditioned on the process never having observed a tree with more than m ϵ-balance edges. However, observing a large number of balance edges in a single tree is an extremely rare event: even in 2 billion proposals in Pennsylvania, there were at most 23 balance edges at $\epsilon = .01$ (observed 1 time). In Virginia, there were at most 18 balance edges at $\epsilon = .01$ (observed 4 times) in 2 billion proposals. Because the bound m was set well higher than this, the effect on any conclusions reached because of this subtle conditioning should be negligible.

state	run type	proposals	adjacency	balance	$1/m$	seam length	accept
VA	`ReCom-A`	1,000,002	–	218,721 *78.1% rej*	–	–	21.9%
VA	`ReCom-B`	1,000,005	304,053 *69.6% rej*	90,950 *70.1% rej*	–	–	9.09%
VA	`ReCom-C`	1,000,005	–	225,040 *77.5% rej*	–	–	22.5%
VA	`ReCom-D`	1,000,011	305,604 *69.4% rej*	90,145 *70.5% rej*	–	–	9.01%
VA	`RevReCom`	1,000,902	301,205 *69.9% rej*	76,009 *74.8% rej*	2524 *96.7% rej*	578 *77.1% rej*	0.0577%
PA	`ReCom-A`	1,000,001	–	246,928 *75.3% rej*	–	–	24.7%
PA	`ReCom-B`	1,000,021	224,804 *77.5% rej*	73,159 *67.5% rej*	–	–	7.32%
PA	`ReCom-C`	1,000,002	–	250,931 *74.9% rej*	–	–	25.1%
PA	`ReCom-D`	1,000,042	227,172 *77.3% rej*	72,225 *68.2% rej*	–	–	7.22%
PA	`RevReCom`	1,000,482	217,050 *78.3% rej*	66,949 *69.2% rej*	2174 *96.8% rej*	443 *79.6% rej*	0.0443%
7x7	`ReCom-A`	1,000,001	–	532,692 *46.7% rej*	–	–	53.3%
7x7	`ReCom-B`	1,000,004	467,867 *53.2% rej*	289,705 *38.1% rej*	–	–	29%
7x7	`ReCom-C`	1,000,001	–	542,793 *45.7% rej*	–	–	54.3%
7x7	`ReCom-D`	1,000,001	469,550 *53.0% rej*	293,290 *37.5% rej*	–	–	29.3%
7x7	`RevReCom`	1,000,008	465,932 *53.4% rej*	292,039 *37.3% rej*	–	164,252 *43.8% rej*	16.4%

Table 2: **Acceptance rates**. Run statistics, showing the number of proposals accepted at each stage, for recombination runs on Virginia (precincts, $k = 11$, $\epsilon = .01$, $m = 30$), Pennsylvania (VTDs, $k = 18$, $\epsilon = .01$, $m = 30$), and a 7×7 grid ($k = 7, \epsilon = 0$, $m = 1$). Roughly 1 million steps are initially proposed in each trial; irregular numbers are due to the batching technique, as described in §C.1. The $1/m$ column is estimated with a geometric variable since it is integrated in the seam length step, as implemented. One observation is that the variants using MST (A and C) are less likely to find a balance edge than the others, which use UST. This is expected behavior, since MST trees are weighted toward having higher-degree vertices, making them harder to split in a balanced fashion.

We sketch the batching strategy here. The low acceptance rate of `RevReCom` on real-world graphs (see Table 2) enables us to secure performance gains by using multiple cores. As a general matter, Markov chains resist parallelization because, by definition, the next step is probabilistically determined by the current state, so a single random walker must consider the proposals at each step. However, if the great majority of proposals are being rejected in a given chain, then we can get an efficiency boost by using parallel workers at the proposal stage that collectively consider a few hundred proposals simultaneously. They can then be ordered randomly and the accepted proposal with lowest index can be passed back to the main thread, at a potentially significant time savings. This is an instance of what is sometimes called "speculative execution"—work is performed that may not be used, in order

to minimize lag time. We have found work by Brockwell from 2006 proposing a similar batching strategy, where he uses the term "pre-fetching" for the partial parallelization [Bro06]. The batch size should be large enough for the advantages to surpass the cost of synchronization overheads, but without leading to many wasted samples. Empirically, we find that a batching strategy is effective for reducing wall-clock compute time on full-scale redistricting problems.

Python GerryChain loads shapefiles or graphs as its input format; the Rust implementation loads graphs with integer-valued attributes. Very long `RevReCom` runs can save selected summary statistics straight to JSONL files. Generally, outputs are available in various serialization formats, including a highly compressed file format called BEN (`gerrytools.readthedocs.io/en/latest/user/ben`).

C.2 Testing algorithms of other authors

To test Forest `ReCom` and SMC, we rely on the implementations provided by their originators, found at [Her] and [KMFI]. Forest `ReCom` is written in Python and Julia while SMC uses R and C++. (We note that other implementations exist as well; for instance, the Redist repo that is the primary home for SMC also contains an R implementation of Forest `ReCom`.) Each codebase has its own ways of loading data and storing outputs, but our replication repo includes helper functions to make inputs and outputs interoperable. Replication materials can be found in [RR].

SMC keeps a great deal of graph data in memory during a run, causing significant RAM consumption (see Table 3); only after a batch run is done are plans written to disk.

We have made a serious effort to collect and present samples by each of the methods in the manner intended by the authors. Typically, Markov chain authors use multistart and enlargement tests to provide convergence heuristics (i.e., comparing runs from multiple starting points, and checking to see that much longer runs perform similarly to the reported runs). This is what we do here, rather than using subsampling and/or the smallest acceptable ensemble sizes to compete for the appearance of efficiency. (We have confirmed that the use of burn-in and subsampling would not materially alter any findings we present.) For Markov chain methods, there is a natural rule of thumb available to choose a sample size for a given accuracy target. A user who desires a sample within $d_{\text{Wass}} < \tau$ of the stationary distribution, for some threshold τ, should at a minimum run chains from different starting conditions until the time-t ensembles are within 2τ of each other.

The SMC authors advocate for the use of the Gelman-Rubin convergence diagnostic $\widehat{R}$, which compares within-sample variance to between-sample variance, usually referring to "our heuristic convergence check that $\widehat{R} \leq 1.05$" [MI23]. Following this scheme, if it tends to be the case that two samples with $S = S_0$ plans pass this test, then we treat S_0 as an adequate batch size to get an accurate sample. Information on the $\widehat{R}$ diagnostic can be found at [GR92, VGS+21].

C.3 Timing comparisons

MCMC Method	Notes	Accepted	Time	Rate
Python `ReCom-A`	1 core	1000 plans	51.56 sec	19.4 accepted/sec
Rust `ReCom-A`	1 core	485,987 plans	107.79 sec	4509 accepted/sec
Rust `ReCom-A`	4 cores	485,983 plans	71.08 sec	6837 accepted/sec
Rust `ReCom-A`	4 cores	487,239 plans	89.68 sec	5443 accepted/sec
Rust `RevReCom`	1 core	2136 plans	43.791 sec	48.8 accepted/sec
Rust `RevReCom`	4 cores, no batching	2033 plans	25.537 sec	79.6 accepted/sec
Rust `RevReCom`	10 cores, batches of 12	2022 plans	5.928 sec	341.1 accepted/sec
Rust `RevReCom`	10 cores, batches of 16	1715 plans	5.401 sec	317.5 accepted/sec
Rust `RevReCom`	4 cores, batches of 32	2105 plans	9.331 sec	225.6 accepted/sec
Forest `ReCom`		2259 plans	682.24 sec	3.3 accepted/sec

SMC batch size (S)	Memory (RAM)	Time	Rate
5000 plans	0.67 GB	107 sec	46.7 produced/sec
20,000 plans	1.76 GB	354 sec	56.5 produced/sec
100,000 plans	10.54 GB	1947 sec	51.4 produced/sec

Table 3: **Basic timing comparison for MCMC methods and SMC.** We include selected timing information to facilitate apples-to-apples comparisons between implementations. Timings are variable, so these single-run figures are intended to be illustrative. We emphasize that speed is not related to sample quality except that practical applications will need runtimes to be on the scale of hours or days, not weeks. For all benchmarking runs, we use the Virginia precinct dual graph with the CD12 seed (Congressional districts from 2012), $\epsilon = .05$, and minimal updaters. We set $m = 30$ for `RevReCom` and SMC. The runs were conducted on an Apple M1 Pro laptop (10 cores) with 16 GB RAM. Python runs were executed in version 3.10.2. SMC runs performed in a Linux virtual machine with 5 cores on the same hardware.

D Empirics

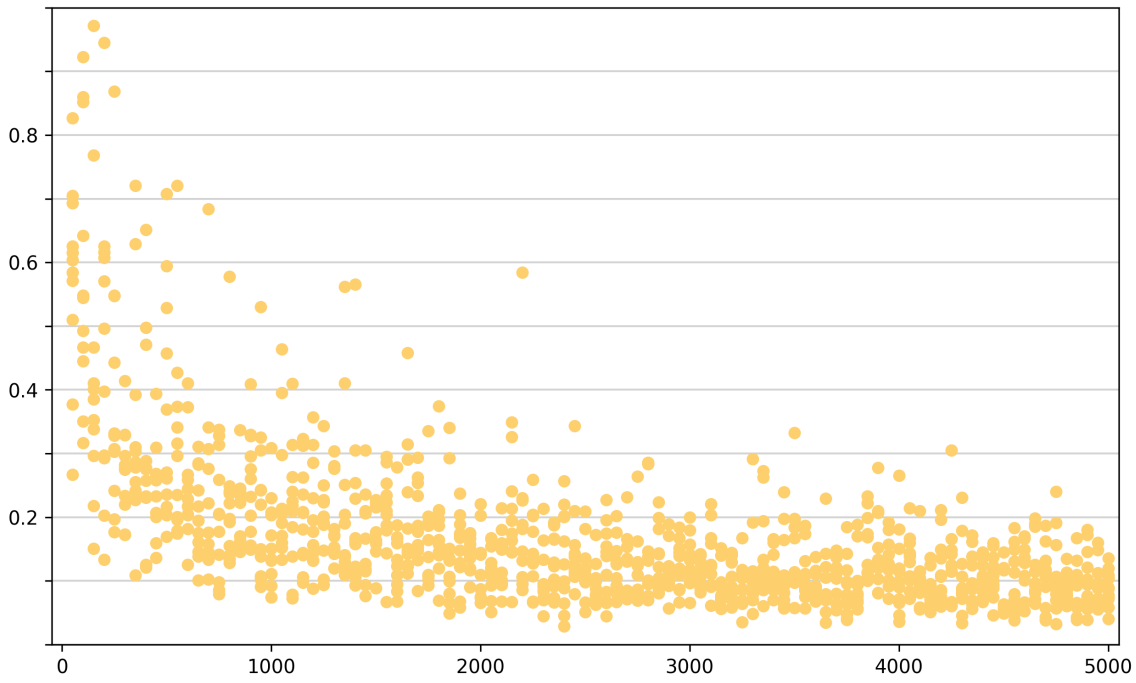
D.1 Data and methods for comparisons on Pennsylvania, Virginia, and the 7×7 grid

Data. The 158,753,814 configurations in the $7 \times 7 \to 7$ districting problem were found using a tool called `enumpart` that was created by Kosuke Imai's research team and can be found in the same Redist repo as the SMC code [KMFI]. The plans were then re-weighted by their spanning tree counts to produce the dataset used in §4.2. Supporting code is available in the replication repo for this paper [RR]. We note that the $7 \times 7 \to 7$ state space was confirmed to be connected by a brute-force computational check.

The maps assessed above in Pennsylvania include the Legislative proposals from 2011 and 2018 and the court's Remedial plan from 2018. The "8th Grade Class" map was created by students of Jon Kimmel at Westtown School in Chester, PA, and was covered by the local press (*How difficult is it to redistrict Pennsylvania? 'Not very,' say area eighth-graders*, available from `whyy.org`). In Virginia, we have used the enacted Congressional maps from 2012 and 2016 as seed plans in Figure 5. In addition, precinct-level Presidential returns from 2016 and U.S. Senate returns from 2016 were used to study partisan outcomes across ensembles of alternative plans. Generally both plans and election data for these time periods are publicly available from Redistricting Data Hub (`redistrictingdatahub.org`). The specific electoral datasets used here were cleaned and prepared by the members of the MGGG Redistricting Lab for research use and are also shared in [RR].

Methods: Choice of election data. For Figure 1 (top), we use the vote counts by precinct for Clinton (Democratic) and Trump (Republican), and for McGinty (Democratic) and Toomey (Republican)—the major-party candidates in the 2016 contests for President and U.S. Senate, respectively. The number of Democratic seats is simply the count of districts in which the Democrat received more votes than the Republican, obtained by summing over the precincts. We note that this is a standard approach: we use statewide (exogenous) elections rather than Congressional (endogenous) elections for the underlying vote pattern in an ensemble analysis, so that moving the district lines does not run up against issues from uncontested districts, variable incumbency advantage, and other district-specific confounding variables. The use of statewide elections in serial rather than composite ensures that the D/R votes draw from "naturalistically" observed patterns; one can still interpret individual contests and be mindful of their anomalies. Though it is standard, this approach does have detractors in political science, who prefer to use regression techniques to blend endogenous races together into a synthetic election while controlling for various confounding factors.

D.2 Detailed SMC diagnostics



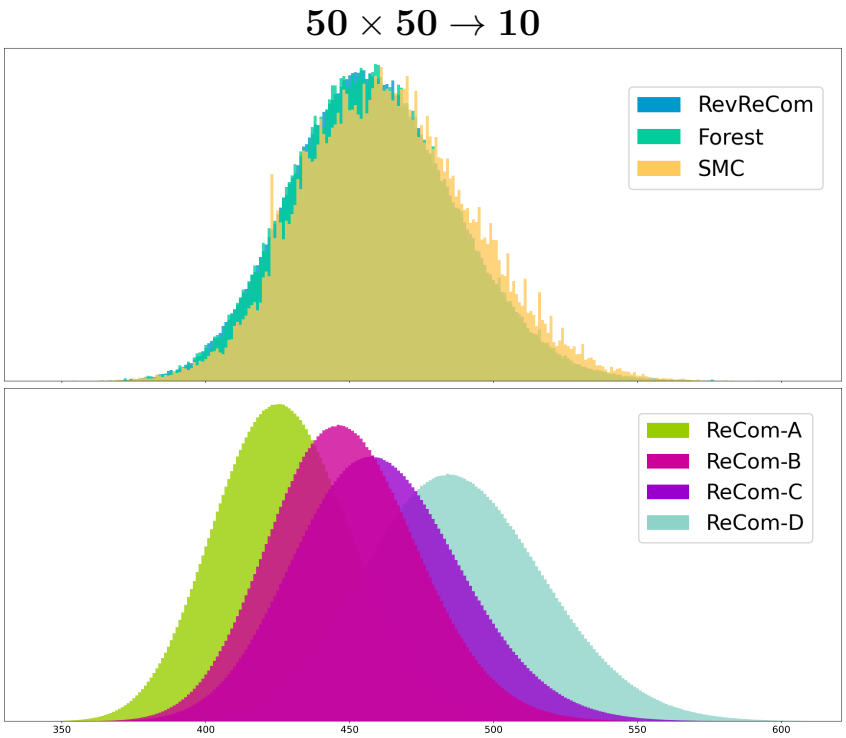
S	50	100	150	200	250	300	350	400	450	500	550	600
# pairs w/ $\widehat{R} < 1.05$ out of 55	40	37	43	37	48	54	49	54	55	53	54	55

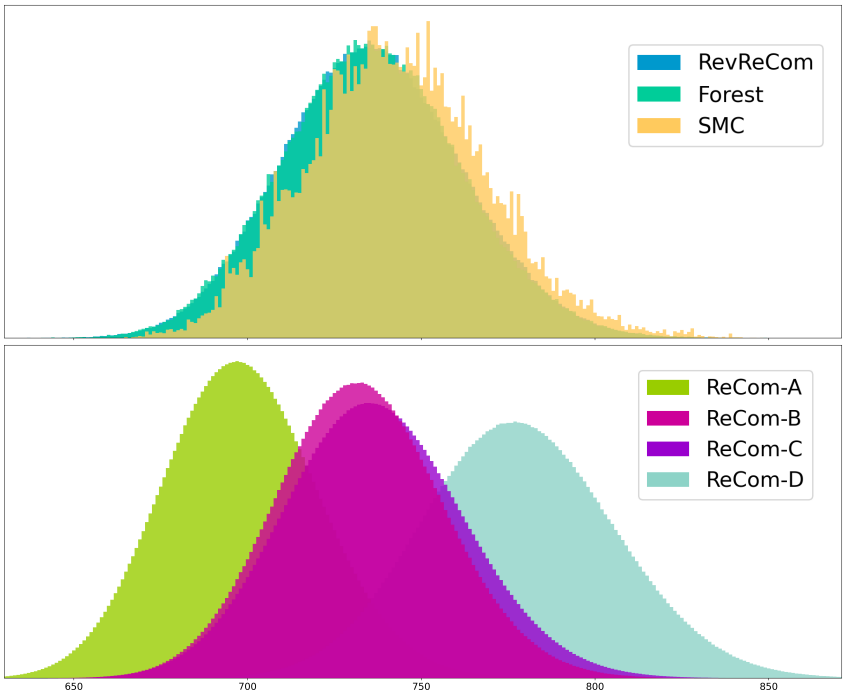
Figure 8: **Detailed SMC diagnostics.** TOP: The scatterplot shows the Wasserstein distance of the sampled cut edge distribution from the π-weighted ground truth for small batches of SMC plans for the 7×7 problem ($S = 50, 100, 150, \ldots, 5000$). Batch sizes up to $S = 100,000$ are shown in Figure 4. BOTTOM: For 11 runs at each each S value, we calculate $\widehat{R}$ for the cut edges statistic for all $\binom{11}{2} = 55$ pairs of runs and count how many of them pass the $\widehat{R} < 1.05$ test pairwise.

The SMC authors use the Gelman–Rubin $\widehat{R}$ statistic, as their main convergence diagnostic. Though better known for use with MCMC methods, it compares within-sample to between-sample variability, so can be used in this setting. More unusually, they designate a particular threshold of $\widehat{R} \leq 1.05$ for practical use in redistricting problems, as cited in §C.2. The statistics shown in Figure 8 illustrate that a sample size of $S = 400$ would be deemed adequately converged by this standard. This produces accuracy similar to the heuristically targeted methods `ReCom`-A,B,C in this problem (see Figure 4).

D.3 50×50 experiments

Finally, we compare `ReCom`-A,B,C,D, `RevReCom`, and SMC samples for the problem of dividing a 50×50 grid into 10, 25, and 50 districts.





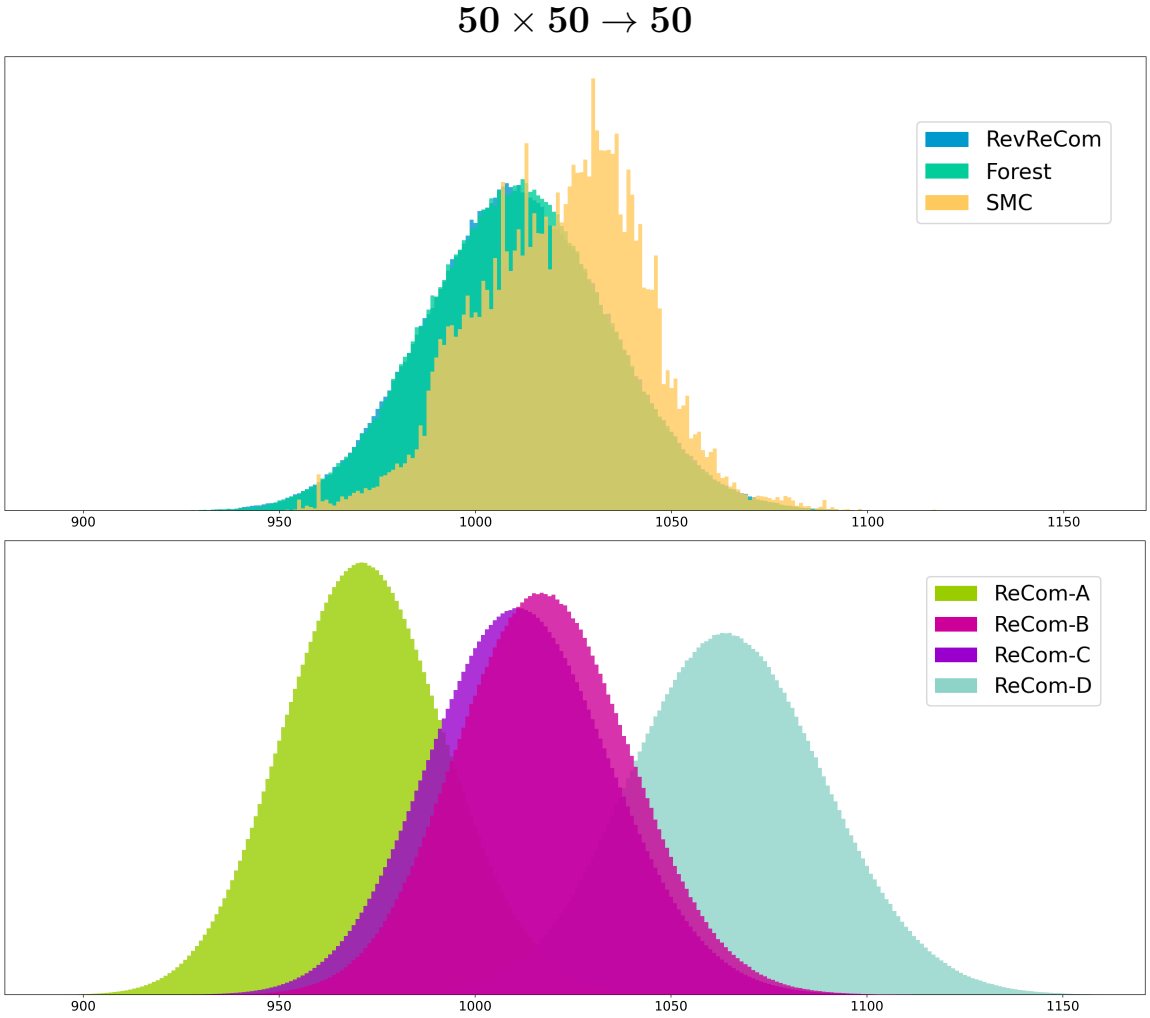


Figure 9: 50×50 **grid comparisons.**

All methods are shown at their largest practical sample sizes (see §2.2). The three methods that provably target the spanning tree distribution π are shown as the top plot in each pair, and the Markov chain methods (`RevReCom` and Forest `ReCom`) give visually indistinguishable results. Since SMC sample quality improves with batch size, we use π-targeted runs with $S = 100,000$ for the best possible results. (Published SMC results usually rely on much smaller batch sizes, even for state-level redistricting.) At this size, the SMC technique is able to get a sample that reasonably resembles the recombination methods when dividing the 50×50 grid into $k = 10$ districts, but its performance suffers, even pushed to the largest achievable sample size, with more districts. The performance of the recombination samplers does not show signs of degrading as the number of districts increases. We note that the size of this test problem—2500 units of a 50×50 grid, and ten to 50 districts—is realistic for real-world settings. For instance, Pennsylvania has about 9000 precincts, $k = 18$ or now $k = 17$ Congressional districts, $k = 50$ state Senate districts, and $k = 203$ state House districts. In Virginia, there are about 5000 precincts, and the number of districts is $k = 11, 40, 100$, respectively.

Consistent with the discussion throughout this paper, the four methods `ReCom`-A,B,C,D all look extremely well converged, but to a steady state that is not identical with π. `ReCom`-C resembles π closely in each example here. Finally, we remark that, going beyond convergence considerations, large samples will still be needed in all methods for estimating finely-binned or high-dimensional statistics.