Characterizing and Generating Redistricting State Dual Graphs

Rhea Acharya, Iñaki Arango, Jeffrey Wang, Lawrence Zhang

April 2024

> "The right of voting for representation is the primary right by which other rights are protected."
> Thomas Paine, *Dissertation on the First Principles of Government*

1 Introduction and Motivation

From the Athenian Academy to the genesis of the American Experiment, voting has long held a central place in the banister of democratic values. Because it serves as a bulwark against centralized power, however, efforts to undermine the electoral process are as old as democracy itself. In the United States, gerrymandering stands out as a particularly egregious example of disenfranchisement. The term, originally coined out of anger after a salamander-shaped district was created during Elbridge Gerry's governorship of Massachusetts in 1812, describes the practice of manipulating electoral district boundaries to favor one party. In addressing the practice, modern computation is both a blessing and a curse. While computational approaches can be (and have been) leveraged to create maps for maximal political advantage [2, 9], they also offer a compelling opportunity to audit proposals and create fairer maps.

In practice, creating districts requires dividing a given piece of land into regions with certain desiderata in mind; examples include equal population between districts, compactness (avoiding long, strung-out districts), connectivity (districts are contiguous), and others. In practice, map-makers seeking to draw N districts begin with some "base" tiling/subdivision, like census blocks or census tracts, and then "glue" these tiles together to form N final pieces that respect the desiderata they have in mind. For computer scientists, this reduces to a problem of *graph partitioning*; mechanically, this means taking any base tiling, creating a node for every tile, and connecting any two nodes with an undirected edge if there exists a positive length boundary between them. Districts can be formed by taking partitions of this base graph; see Figure 1 for an illustration of this process. On these dual graphs, algorithms based on MCMC [6, 8], spanning trees [3], and sequential MC [13] have proven fruitful for creating partitions that correspond to actual districts which satisfy the aforementioned desiderata.

Starting in the early 2010s, expert witnesses in gerrymandering cases began using these computational methods as evidence in court. To do so, they compared a given set of legislature-approved districts to an *ensemble* of districts created from an algorithm. Many of these cases have revolved around a common line of reasoning: if the algorithm could be shown to sample uniformly from the space of all possible maps and the given set of districts performs poorly in a large cohort of other possible maps, it must be deemed unfair.

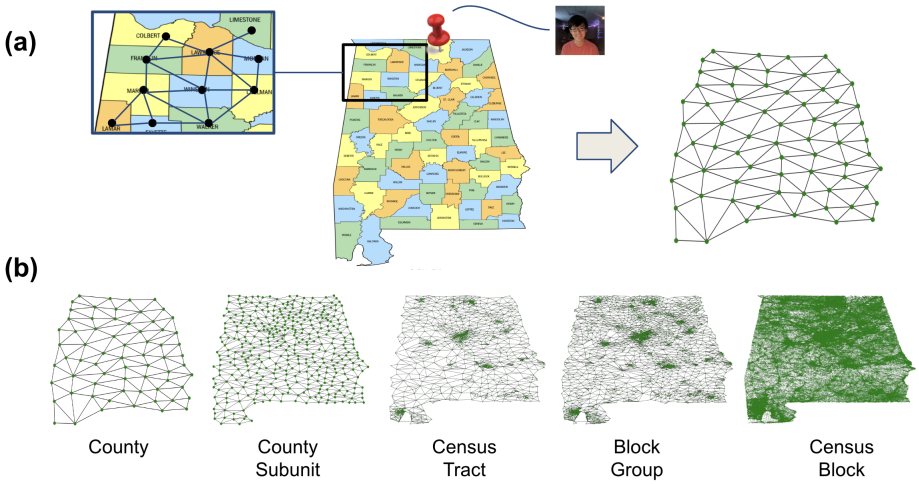

Figure 1: **(a)** Beginning from the tiling Alabama's counties, we illustrate the process of creating the dual graph. (Madison County is where our team member, Lawrence, is from.) **(b)** We illustrate all five levels of data we consider, reflected here as all five levels of Alabama's dual graph. From a dual graph, any graph partioning algorithm can provide a set of partitions that correspond to a "bundling" of the original tiles, which represents a set of districts. Graphs are provided by [5].

While ensuring these algorithms are performant is critical, there is a relatively small pool of baselines to test; in practice, studies typically examine algorithm performance on a rectangular lattice dual, as well as a few states. To empirically validate these districting algorithms, then, a critical question arises:

> What do real-life district maps look like, and how can we characterize/generate them?

In this work, we seek to answer this question. First, we observe that dual graphs are quite different from rectangular lattices or other standard graphs in network theory. In particular, the base dual graphs on which districting algorithms are run are almost triangular and almost planar ("ATAP"). Specifically, they are mostly planar (or planar barring a few offending nodes with their incident edges), and for the planar graphs, most faces are triangular.[1] Separate from benchmarking redistricting algorithms, characterizing these properties may also of theoretical interest because many graph algorithms run faster on graphs with certain properties (e.g. planarity) [11]. We then create null models that provably generate these ATAP graphs, empirically optimize parameters, and compare resulting benchmark graph statistics with real state dual graphs with a custom-built robust benchmarking library located here.

2 Background

To begin, we introduce some background and related work that undergirds our algorithms and empirical analysis.

[1] Note that a triangular face corresponds to any connected set of three tiles. A quadrilateral face corresponds to four tiles meeting at a point, and so on.

Definition 1. *(Planarity) A planar graph has some planar embedding where all edges can be drawn without crossing. Planar graphs divide the plane into regions called* ***faces****.*

Faces are only well-defined on a planar graph, so we adapt the following definition of triangularity:

Definition 2. *(Triangularity) A pure triangular graph is a planar graph that has faces which are exclusively triangles.*

We use a variety of graph statistics in our experiments later as well.

Definition 3. *(Assortativity coefficient) The assortativity coefficient of a graph is the correlation of degree between pairs of linked nodes.*

Definition 4. *(Radius) The radius of a graph is the minimum* ***eccentricity*** *of all nodes in the graph, where each node's eccentricity is the maximum distance between itself and any other node.*

Definition 5. *(Diameter) The diameter of the graph is the maximum eccentricity out of all the nodes in the graph.*

Definition 6. *(Clustering coefficient) The clustering coefficient of a graph G is the fraction of all possible triangles completed and present in the graph, i.e.,* $\frac{3\cdot \textit{number of triangles in } G}{\textit{number of triads in } G}$*, where a triad in G is a single vertex in G with edges to an unordered pair of other vertices.*

Note that radius and diameter are only well-defined for connected graphs.

To formalize a notion of "almost-planarity," we consider characterizations and properties of planar graphs.

Theorem 1. *(Kuratowski): A graph is nonplanar if and only if it contain some subgraph that is a subdivision isomorphic to K5 or K3,3.*

Theorem 2. *(Euler's Formula): For a finite, connected planar graph with n nodes, e edges, and f faces (in any planar embedding of the graph),*

$$v - e + f = 2$$

A variety of other characterizations exist, but these are the ones we will utilize later.

In considering notions almost-planarity, we run into a bevy of NP-completeness results. One natural notion is to define the "almost" in terms of the minimal number of edges or vertices to remove until reaching planarity. Unfortunately, these are both NP-complete problems, although a 4/9-approximation algorithm exists for the maximal planar subgraph problem, as well as a $\mathcal{O}(2^{k\log k})\cdot N$ algorithm for determining if a N-node graph G has some subset of vertices S where $|S| = k$ and $G \setminus S$ is planar [10, 4]. While this wall of NP-completeness may be discouraging, it turns out that in our setting of state dual graphs for redistricting, we can quite simply bound the minimal number of vertices to remove until planarity.

3 Examining Real State Dual Graphs and Formulating ATAP

We first examine all 50 real state dual graphs at the Census tract level for apparent properties to motivate our theoretical definition of ATAP in Figure 2, where we plot the distributions of a broad variety of graph

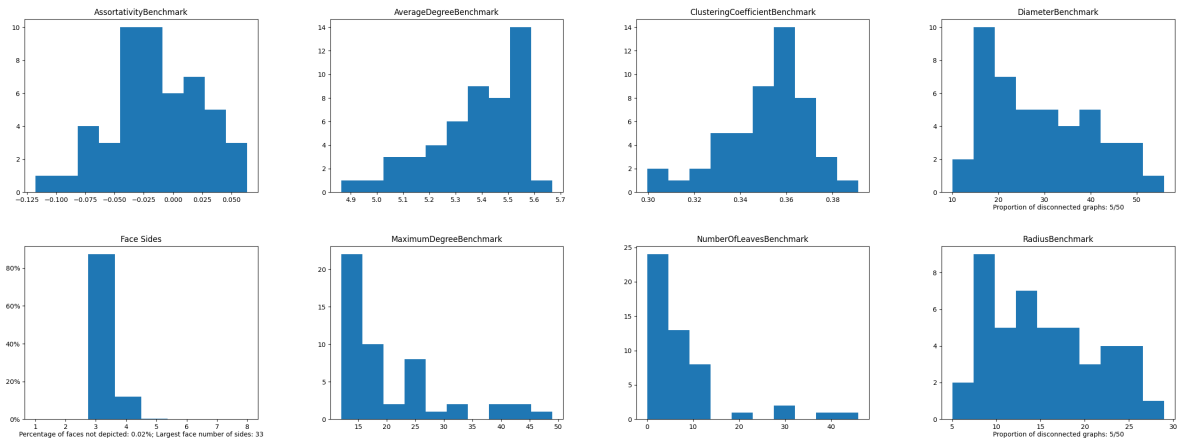

Figure 2: Real Census Tract State Dual Graph Statistics. Note that the statistics here that require certain graph properties (e.g. faces require planarity) only reflect the distribution over states that have that property.

statistics across all the states. Similar visualizations and conclusions can be found in Appendix 8.4 for state dual graphs at the remaining Census levels.

First, we note that the large majority of state maps are connected, as seen in the radius and diameter histogram footnotes. Then, the most visible, obvious characteristic of real state dual graphs is their almost triangular nature, as evidenced by the large majority of faces being triangular, average degree being slightly under six, low clustering coefficients and maximum degree, all of which is what we would expect for graphs that loosely resemble a triangular tessellation of the plane. More subtly, the state dual graphs are almost planar, in the sense that most dual graphs are planar as seen above. Importantly, the remaining dual graphs are only a few vertices away from being planar as well, as non-planarity in these dual graphs specifically used for districting applications arises only from split base tiles, as without split base tiles, the dual graph of a planar district map is clearly planar. For instance, Louisiana is the only state with a non-planar county dual, as St. Martin's Parish is disconnected (see Appendix 8.1). Hence, an upper bound on the number of vertices to remove from the dual graph until planarity, which we deem to be offending nodes, is simply the the number of disconnected base tiles we have, which is a property of the underlying data that we can measure. Therefore, to accomodate this, our definition of ATAP is is loose in the sense that it is relatively easy to "generate" a graph that satisfies our definition, but hard to verify.

To see this almost planarity and almost triangularity at the Census tract level, for instance, 39 out of 50 state dual graphs are planar, with overwhelmingly triangular faces (see Appendix 8.6 for face distributions state-by-state across different Census levels). Additionally, by examining the raw Census data shapefiles and specifically the number of parts in each polygon corresponding to a district (a hefty task!), we empirically find that all 11 states with non-planar dual graphs at the Census tract level have under 1% of tracts that are split, with most states having just a few basis points of offending nodes: for example, California has 8057 total tracts and only 0.11% have potential to be split (see Appendix 8.2 for more statistics at all Census levels). In sum, all of these observations motivate the following natural definition of almost-triangular, almost-planar graphs, capturing the essence of real state dual graphs.

Definition 7. *(ϵ-δ ATAP) For $0 \leq \epsilon, \delta \leq 1$, graph $G = (V, E)$ is ϵ-δ almost-triangular-almost-planar (ATAP) if $\exists S \subseteq V$ with $|S| \leq \epsilon |V|$ such that G', the subgraph induced in G by S (i.e., G minus the vertices in S and*

their incident edges), is planar and has a planar embedding that satisfies

$$\frac{\textit{number of triangular faces in } G'}{\textit{total number of faces in } G'} \geq 1 - \delta$$

Intuitively, ϵ describes how planar the graph is, and δ describes how triangular the graph is, with smaller parameters corresponding to the graph being closer to planarity or triangularity. Note that the definition is loose in the sense that we let $|S|$ be *at most* $\epsilon|V|$ and the proportion of non-triangular faces be *at most* δ, so that if a graph is $\epsilon - \delta$ ATAP, then it is also $\epsilon' - \delta'$ ATAP, for $\epsilon' \geq \epsilon$ and $\delta' \geq \delta$. In particular, *any* graph is $1 - \delta$ ATAP for any δ (since you can just trivially take $S = V$), and graphs that are $0 - 0$ ATAP are pure triangular graphs. Finally, we note that while the number of faces in a planar graph is invariant to the specific planar embedding chosen, the distribution of the number of sides of the faces is not invariant, so our definition again uses a weak "there exists" quantifier for a nearly-triangular planar embedding of a particular graph, just like we only require the existence of one such $S \subseteq V$ to be removed from graph to recover planarity (intuitively, the offending nodes corresponding to the split districts in the dual graph).

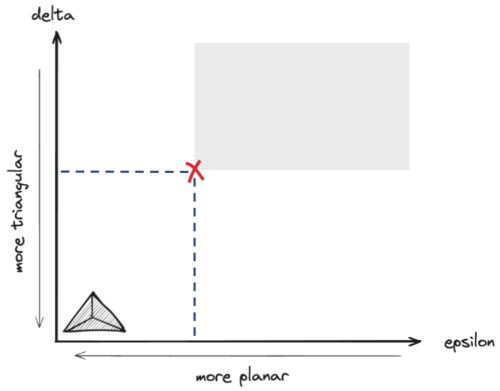

Figure 3: Visualization of $\epsilon - \delta$ ATAP definition properties

4 Null Models

We consider the following broad classes of parameterizable null models that attempt to create graphs that overall match the distribution of real state dual graphs well, under the theoretical framework and empirical properties of creating almost triangular, almost planar graphs.

4.1 Triangulations

Algorithm. Plot n points uniformly at random in the unit square. Then, create the Delaunay triangulation of those points, which subdivides their convex hull into triangles whose circumcircles do not contain any of the points [7]. For our purposes, we mostly abstract the details of the Delaunay triangulation and use it simply to create a graph $G = (V, E)$ where all of the internal faces are triangles. The resulting graph is therefore completely planar and also almost completely triangular (barring the outer face).

To modify G, we randomly remove edges from G to create a graph G', by removing exactly $k = p|E|$ edges randomly chosen from the graph for some tunable parameter `p_delete`, representing the probability any single edge is removed. Notably, however, we do not allow this edge removal process to disconnect the graph. Finally, to make G' almost planar, we can use the following perturbation idea: simply add an appropriate number m of offending nodes to produce the final graph G''. For each offending node, we iteratively connect it to random nodes in G' until planarity is violated. Intuitively, it is clear that G'' is almost triangular and almost planar and is indeed non-trivially $\epsilon-\delta$ ATAP as well. Note that this method, by definition, produces only connected graphs too.

From a practical perspective, the Delaunay triangulation of n points can be computed in $O(nlogn)$ time using a divide and conquer algorithm with a clever $O(n)$ merging step [12]. Randomly removing $k = p|E|$ edges while ensuring continuity takes $O(n^2)$ time (as seen in the proof of the below theorem, $O(|E|) = O(n)$, and we can verify each potential removal does not disconnect the graph with a linear-time traversal of the graph). Finally, the perturbation runs in worst case $O(\frac{\epsilon}{1-\epsilon}n^3)$ time. Supposing a null model adds m offending nodes to an n-node graph, we have $\epsilon = \frac{m}{n+m} \implies m = \frac{\epsilon}{1-\epsilon}n$. For each of those offending nodes, we randomly add edges to at most n nodes and thus check planarity in $O(n)$ time at most n times for a total of $O(n^2)$. Hence, this null model is readily deployable and runs in total in polynomial time!

Theorem 3. *(Triangulation null model produces non-trivially ATAP graphs) Consider the triangulation null model that produces a n-vertex initial triangulation G, removes k randomly chosen edges to produce G', and then adds m offending nodes to produce the final graph G''. Then, for $2n-1-b-k > 0$, G'' is $\epsilon-\delta$ ATAP, for $\epsilon = \frac{m}{n+m}$ and $\delta = \frac{k+1}{2n-1-b-k}$, where b is the number of vertices on the convex hull of the initial triangulation.*

Proof. Note that, by the process that creates the offending nodes, m is an upper bound on the number of nodes that need to be removed for G'' to be planar, i.e., we can remove $\epsilon = \frac{m}{n+m}$ fraction of the nodes to recover planarity and return to G'. To find δ, we now lower bound the number of faces that are triangular in G'. Note that each edge of the initial triangulation G touches the two faces to which the edge is adjacent to, and G has all triangular faces but the outside face, which is formed by the convex hull and is b-sided. Hence, $2e = 3(f-1)+b$, where e is the number of edges and f is the number of faces in G. By Euler's formula, $n-e+f = 2$, so solving for f gives $f = 2n-1-b$, meaning G has $2n-2-b$ initial triangles. Removing k edges from G removes at most $2k$ of these triangles, again because each edge touches only two faces, so there are at least $2n-2-b-2k$ triangles in G'. By Euler's formula again, since G' remains connected by definition of the null model and the number of vertices stays constant, the number of edges decreasing by k implies that there are $f-k = 2n-1-b-k$ faces in G'. Hence,

$$\frac{\text{number of triangular faces in } G'}{\text{total number of faces in } G'} \geq \frac{2n-2-b-2k}{2n-1-b-k} \geq 1-\delta$$

and solving gives $\delta \geq \frac{k+1}{2n-1-b-k}$, so G'' is indeed $\epsilon-\delta$ ATAP for the claimed values of ϵ and δ. □

As was clear by this null model starting from a triangulation of the nodes, but formally verified above, we thus have confidence that this null model generates almost triangular, almost planar connected graphs. Note that this model has a tunable parameter p.

4.2 Random Walks

Algorithm. Starting with a rectangular grid graph (visualizing as a rectangle of tiles where each tile is a node that has edges to adjacent tiles), we initialize n agents, where n is desired number of final nodes in the dual graph, randomly placing each of these n agents on their own starting square in the grid, colored according to the agent. At each time step, in a random order, each agent will move to a randomly chosen, uncolored square adjacent to *any* of its own colored squares (not just adjacent to the current square it is at), staying in place if all adjacent squares are already colored. Once all of the squares have been colored, we have generated a tiling of the original large rectangle into differently colored contiguous regions, which we convert to the corresponding dual graph. Note that this dual graph is planar, since the original tiling we produce is planar.[2]

Note this algorithm always terminates because if there are still uncolored squares, there must be at least one such uncolored square adjacent to a colored square (otherwise the whole grid would be uncolored), and the corresponding agent can therefore color that adjacent uncolored square. To introduce ϵ almost planarity to the dual graph, we can employ the same perturbation idea mentioned previously, thus overall producing an almost planar and connected graph. Finally, note that this algorithm runs in polynomial time since every walker will choose a possible direction to move at random, sequentially, until all squares are filled; sampling a direction (or removing the agent when it can no longer move) takes constant time so the runtime is polynomial to the size of the grid. For our purposes, we set it such that for n agents the square grid had $5n$ tiles.

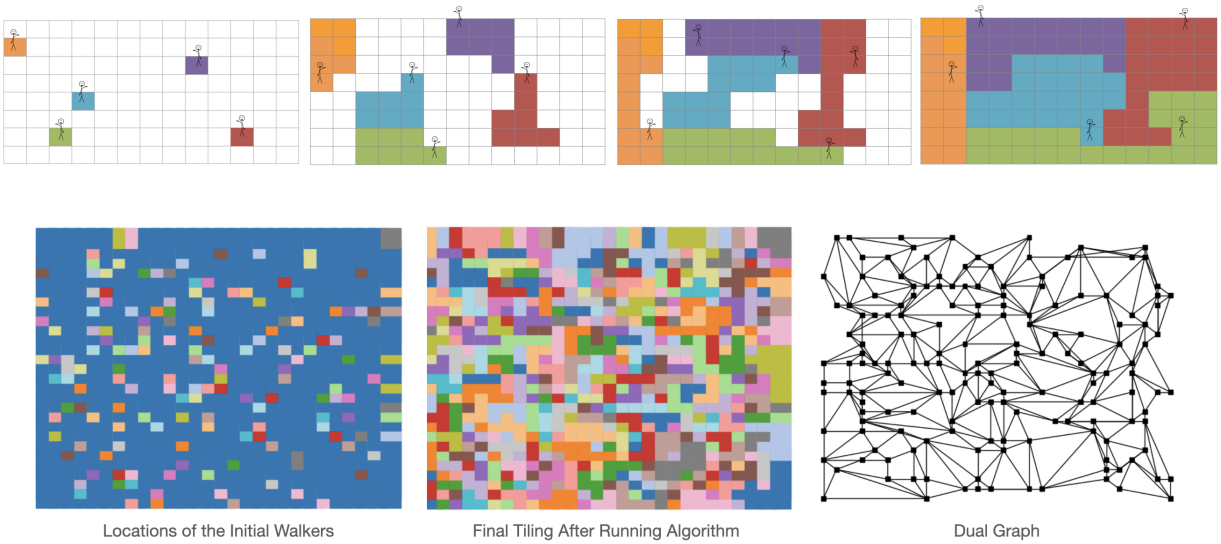

Figure 4: **Top**: A random walk at various points in time, generating an example tiling. The dual of this tiling will be the base dual of a redistricting algorithm. **Bottom**: The algorithm run in practice. We begin with initial walkers, get a tiling after running the algorithm, and generate a dual (note that the dual here has some crossing edges, but this is an artifact of the visualization software).

Motivation. Though we lack a formal δ almost triangular proof, we have concrete intuition by relating

[2]We essentially do random walks on a *very* dense grid to get tiles corresponding to a dual graph that we want to run our actual redistricting algorithm on.

this random walk null model to the triangulation null model above. It turns out that the Delaunay triangulation is the dual graph of the Voronoi diagram for the n original points in the triangulation null model [7]. The Voronoi diagram partitions the plane into n regions that each contain all points in the plane closest to the corresponding original point; roughly speaking, the Voronoi diagram for n original points is like “flood-filling" (expanding outward and coloring at a constant rate from all original points) on a very dense rectangular grid (that approximates space itself), whose dual graph we have seen above produces nearly all triangular faces except the exterior face. Then, we can interpret the random walk null model for n points as randomly exploring in varying directions (instead of consistently exploring in all directions as in flood-filling) on a less dense rectangular grid, a similar regime whose dual graph we might expect to also have mostly triangular faces. Thus, in expectation, randomly exploring in varying directions from a point should give similar results as simply exploring in all directions at a constant speed.

One note: since we have no edge deletion step in this random walk null model, and by the rectangular nature of the underlying grid (meaning only at most four differently colored squares can meet at any single intersection), the dual graph generated will only have triangular and rectangular faces, in contrast to our triangulation edge deletion model, where by deleting the right edges randomly we can have arbitrarily large polygons. Note also that this model does not have any tunable parameters.

4.3 Waxman Graphs

Algorithm. Consider the following random geometric graph formulation. Plot n points uniformly at random in the unit square. Then, we construct edges between each pair of nodes x, y with probability $\alpha \cdot e^{\frac{-d(x,y)}{\beta \cdot L}}$, where $d(x,y)$ is the Euclidean distance between the points x, y, L is the maximum distance between any two points, α is a parameter that reflects the desired overall density of final graph (with higher α corresponding to greater density), and β is a parameter related to rate at which probability declines with distance (with a higher β corresponding to a faster rate).

Motivation. Mechanically, the motivation for using Waxman Graphs is that these graphs are likely that there are many compact triangular faces; if a vertex is connected to two other vertices, those vertices are probably close together and thus likely to also be connected. However, we note that the resulting graph produced may be disconnected and non-planar, depending on the specific tunable parameters α and β chosen, just like real state dual graphs.

5 Methods

We now turn to empirically evaluating how well our null models could do in producing dual graphs that are similar to real state duals. One quick note is that as mentioned previously, most states have no offending nodes or and the rest have very few offending nodes (e.g., California has 9 offending nodes at the tract level, see Appendix 8.2), which do not influence the graph-wide statistics we consider above to any significant extent on graphs of thousands of nodes. Thus, when evaluating the triangulation and random walk null models, for sake of simplicity, we do not implement the perturbation to achieve ϵ almost planarity and instead use the planar duals directly generated instead.

5.1 Benchmarking

We use many graph statistics as benchmarks to compare the distribution of state dual graphs at a particular level to the distribution of graphs generated by our null models with particular parameterizations to match a particular level. Specifically, we consider a graph's average degree, maximum degree, number of leaves (i.e., in undirected graphs, nodes with degree 1), clustering coefficient, radius and diameter (defined only on connected graphs, so we only benchmark these on connected state dual graphs and connected graphs generated by our first two null models), and proportion of its faces that are triangular (defined only on planar graphs, so we only benchmark this on planar state dual graphs and planar graphs generated by our first two null models, again without the perturbation applied).

Finally, note that each Census level, the real state dual graph distribution has 50 graphs with different numbers of nodes. To compare our null models faithfully, we also generate 50 graphs each with number of nodes corresponding to a particular state. We do this because we can imagine policymakers and researchers using our null models in the future to generate close-to-life dual graphs of a certain size specified via the number of nodes, an input parameter in all of our null models.

5.2 Parameter Optimization and Evaluation

For our nonparametric random walk model, there are no parameters to optimize or tune, so we simply compute a loss for each benchmark statistic, which captures how "different" the distribution of that statistic is in our null model's graphs versus in real state dual graphs. For our parametric models, the triangulation and Waxman graph models, we optimize parameters for benchmarking by constructing a grid of possible parameter values (varying `p_delete` for the triangulation model, and α and β for Waxman) and computing a loss for each benchmark for each set of parameters. For each benchmark, we keep the set of parameters that minimizes the loss, also computing how much those parameters inflate the losses of the other benchmark statistics compared to their minimum losses, which we display in the tables in our results 6. Intuitively, we do this to display which parameter values, if any, perform relatively well on all or most graph statistics we consider, as displaying the actual loss numbers themselves would not be very interpretable.

To calculate loss [3], we compute the L2 (Euclidean) distance between the vector of graph statistic values from the real state duals and the vector from the null model. Since the null model vector contains for position i the statistic of the graph generated based on the constraints (i.e., number of nodes) of the i-th real life map, and we would like for real and null model generated graphs of the same size to be similar in various statistics, using L2 distance is natural.

6 Results

For sake of brevity, we empirically examine state maps at the Census tract level, formed via subdivisions of state counties, where the dual graphs are complicated and large enough to display some tail behaviors

[3] We first tried calculating losses for each benchmark by computing the Kullback-Leibler divergence $D_{\mathrm{KL}}(P\,||\,Q)$ where P is the distribution of the benchmark statistic for real state dual graphs, and Q is the distribution of the statistic for graphs generated by our null models. This was not fruitful, however, because $D_{\mathrm{KL}}(P\,||\,Q) = \infty$ if there exists an x such that $P(x) > 0$ and $Q(x) = 0$. This can happen at times with models that do not provide the best fits, so we eventually settled for an alternative approach.

but small enough to be computationally feasible.

We first examine our parameterized null models, triangulation and Waxman. First, we focus on the graph statistics well-defined on all graphs (even non-planar and disconnected ones), in particular defined for all 50 state duals and all graphs generated by triangulation and Waxman. Below, each row corresponds to a parameter set (shown in the parameters column) that minimizes loss (value shown in the focus loss column) for a particular graph statistic (shown in the benchmark focus column), with the resulting percentage increases in loss for other non-optimized graph statistics shown in the "Percentage Change with Regards to Optimal" columns. The Waxman null model parameters are listed as α, β.

Model	Benchmark Focus	Parameters	Focus Loss	Percentage Change with Regards to Optimal			
				AD	CC	MD	NOL
`triangulation`	Average Degree	0.12	2.695	0.00%	31.23%	28.38%	18.05%
	Clustering Coefficient	0.019	0.431	63.14%	0.00%	52.75%	14.39%
	Maximum Degree	0.15	7.348	10.79%	72.86%	0.00%	17.37%
	Number of Leaves	0.23	13.71	79.23%	84.83%	21.72%	0.00%
`Waxman`	Average Degree	0.1, 0.6	16.93	0.00%	310.55%	14.66%	221.75%
	Clustering Coefficient	1.0, 0.5	0.6644	1275.65%	0.00%	1100.33%	1.02%
	Maximum Degree	0.1, 0.4	25.02	5.77%	314.80%	0.00%	343.67%
	Number Of Leaves	1.0, 0.9	15.62	1836.18%	158.90%	1438.88%	0.00%

Table 1: Parameter optimization results at the Census tract level based on all 50 state duals, including nonplanar and disconnected graphs (allowing only a subset of all benchmark statistics to be run)

We see in Table 1 that generally, the parameters that optimize for each of the benchmark statistics are different (by a significant degree), for both null models. In particular, when optimizing for a single benchmark statistic, the loss of the other benchmarks (compared to the optimal value attainable for every other benchmark using different parameters) grows significantly, and with high variance. The increases in loss for the other unoptimized benchmarks range from a 1% increase to over a 1000% increase. In general, however, the triangulation model sees less variance, with the optimal parameters being set to relatively low values of p (representing the probability any certain edge is deleted from the original triangulation) when optimizing for any single benchmark, and with loss increases for the other benchmarks under 80%, suggesting it is more reasonable to pick an all-around good value of p that simultaneously matches the real state duals' distributions on all of the above statistics.

Next, we focus on the graph statistics well-defined on planar and connected graphs (including the above, but adding diameter, radius, and the proportion of triangular faces as graph statistics), in particular defined for a subset of the 50 state duals and only the graphs generated by the triangulation null model.

We see in Table 2 that we see similar overall results of optimizing for one benchmark statistic resulting in high, varying increases in loss for the other benchmarks, ranging from 2% to over 1000% as well. We note that it seems particularly hard to match the real state duals' diameter and radius statistics, which makes sense given the simplifying geometric assumptions our null models make about the distributions of points in space (namely, initializing them uniformly at random in the unit square), which does not permit the diversity of state shapes necessarily that we see. It is also difficult to match the proportion of triangular faces seen in real state duals, except when optimizing the clustering coefficient statistic, which is reasonable because they both in some sense measure the proportion of triangles in the graph.

Model	Benchmark Focus	Parameters	Focus Loss	Percentage Change with Regards to Optimal						
				AD	CC	MD	NOL	D	R	PTF
`tri`	Average Degree	0.12	2.155	0.00%	233.42%	11.80%	27.75%	124.77%	71.31%	131.57%
	Clustering Coefficient	0.0	0.1425	120.04%	0.00%	45.44%	22.85%	144.09%	89.39%	10.83%
	Maximum Degree	0.16	7.211	27.77%	309.87%	0.00%	34.07%	111.70%	58.80%	191.23%
	Number Of Leaves	0.18	12.77	49.96%	504.36%	31.56%	0.00%	109.59%	53.93%	225.43%
	Diameter	0.46	13.82	460.54%	1051.19%	100.48%	422.27%	0.00%	31.88%	570.32%
	Radius	0.41	6.782	393.89%	1039.90%	75.96%	254.81%	14.41%	0.00%	570.41%
	Prop. of Triangular Faces	0.027	0.4891	79.04%	125.22%	39.37%	24.09%	141.29%	84.74%	0.00%

Table 2: Parameter optimization results at the Census tract level based on the subset of the state duals that are both connected and planar (allowing all benchmark statistics to be run)

Optimizing with respect to average degree and maximum degree perform the next best in regards to also matching the triangularity of the state duals, which again seems reasonable because, as explored earlier, those statistics tend to be fairly closely associated with and consistent to triangular tessellations.

Therefore, in our analysis of our parametrized models overall (particularly of the triangulation model designed to produce ϵ-δ ATAP graphs), we conclude that while it is possible to satisfy almost triangularity with a specific parametrization, matching the proportion of triangular faces well, the other graph statistics that we might also practically be interested in do not also come for free, with statistics more related to triangularity (e.g., degree statistics and clustering coefficient) coming closer than those unrelated to triangularity (e.g., diameter, radius, and number of leaves). This reflects the prioritization of triangularity (in addition to almost planarity) in our characterization of state duals as ϵ-δ ATAP, and suggests a more flexible, encompassing definition is needed to better match more characteristics of state duals.

We now examine our unparametrized random walk null model. Since we no longer have a loss to minimize via parameter grid search, we instead evaluate the null model's performance by comparing the distribution of statistics it generates for a certain graph size (with n nodes) to a real state dual at the tract level with the same number of nodes, as shown in Appendix 8.3. To do so, we did this ten times: sample 5 random states, randomly instantiate a random walk with the same number of starting agents, and then compare the stats of the resulting duals. We see that the floodfill model is fairly performant, as it closely matches the stats.

7 Conclusion and Future Work

In this paper, we tackle the problem of generating realistic-looking dual graphs for benchmarking redistricting algorithms. We propose a notion of "almost triangularity and almost planarity" and evaluate algorithms for generating such graphs. We create a robust benchmarking library for generating and testing these graphs against the distribution of dual graphs from real states.

We think there are a few natural follow-up questions to this work. First, we'd be excited to extend our benchmarking library to integrate with existing packages for conducting redistricting analysis. It would be very interesting to run [6, 8, 3, 13] and other algorithms on our simulated duals. Next, we think it would be important to add more parameterization to our model, since we note that there are fairly heavy tails on most graph statistics with real state duals, due to geographic or historical quirks. See

Appendix 8.5 for examples of high rectangularity and low average node degree. Our current models typically don't generate such unusual "tail" outcomes with a low number of nodes, although this would not be terribly complicated to do (e.g. by baking in a sparsity parameter into the triangulation null model). Finally, we're also interested in studying graph properties at urban-rural interfaces. One explanation for almost-triangularity, for instance, emerges in the state of Illinois, where most rectangular faces appear in dense urban areas (city blocks) and sparse rural areas (census tracts); this means rectangles should not be generated uniformly in a truly faithful null model. Characterizing these interfaces and integrating information about them into null models is a promising line of future research.

We would like to thank Professor Ariel Procaccia for an amazing semester of learning and discussion in CS 238, and Professors Moon Duchin and Daryl DeFord for their guidance from originating the idea to answering all of our persistent questions.

References

[1] Parish History | St. Martin Clerk — smpcoc2.com. https://www.smpcoc2.com/copy-of-parish-history. [Accessed 27-04-2024].

[2] Robert Brutvan. Illinois' 'extreme' risk of gerrymandering becomes reality through congressional map — illinoispolicy.org. https://www.illinoispolicy.org/illinois-extreme-risk-of-gerrymandering-becomes-reality-through-congressional-map/, 2021.

[3] Sarah Cannon, Moon Duchin, Dana Randall, and Parker Rule. Spanning tree methods for sampling graph partitions, 2022.

[4] Gruia Călinescu, Cristina G. Fernandes, Ulrich Finkler, and Howard J. Karloff. A better approximation algorithm for finding planar subgraphs. *J. Algorithms*, 27:269–302, 1996.

[5] Daryl R. DeFord. Dual Graphs — people.csail.mit.edu. https://people.csail.mit.edu/ddeford/dual_graphs.

[6] Daryl R. DeFord, Moon Duchin, and Justin M. Solomon. Recombination: A family of markov chains for redistricting. *ArXiv*, abs/1911.05725, 2019.

[7] Boris Delaunay. Sur la sphere vide. *Izvestia Akademii Nauk SSSR*, 7:793–800, 1934.

[8] Benjamin Fifield, Michael J. Higgins, Kosuke Imai, and Alexander Tarr. Automated redistricting simulation using markov chain monte carlo. *Journal of Computational and Graphical Statistics*, 29:715 – 728, 2020.

[9] https://www.nytimes.com/by/maggie astor. North Carolina Republicans Approve House Map That Flips at Least Three Seats — nytimes.com. https://www.nytimes.com/2023/10/26/us/politics/north-carolina-republicans-gerrymander.html, 2023.

[10] Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A near-optimal planarization algorithm. In *ACM-SIAM Symposium on Discrete Algorithms*, 2014.

[11] Philip N. Klein, Satish Rao, Monika Henzinger, and Sairam Subramanian. Faster shortest-path algorithms for planar graphs. In *Symposium on the Theory of Computing*, 1994.

[12] Geo Leach. Improving worst-case optimal delaunay triangulation algorithms. In *4th Canadian Conference on Computational Geometry*, volume 2, page 15. Citeseer, 1992.

[13] Cory McCartan and Kosuke Imai. Sequential monte carlo for sampling balanced and compact redistricting plans. *The Annals of Applied Statistics*, 17(4), December 2023.

8 Appendix

8.1 Nonplanar State Dual Graph Example

Louisiana is the only state with a non-planar dual graph at the county level. To see why, examine Figure 5. St. Martin's Parish is discontiguous—historical records from the Parish Clerk [1] illustrate why:

> In 1868, Iberia Parish was formed from parts of St. Martin and St. Mary Parishes. As a result of this separation, compounded by a surveyor's error, St. Martin Parish has two non-contiguous parts.

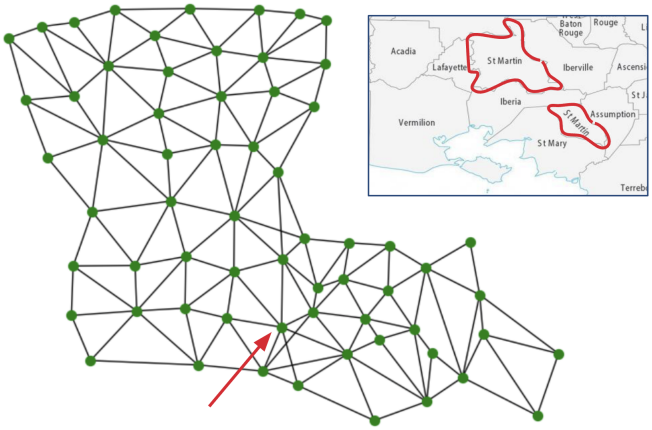

Figure 5: St. Martin's Parish in Louisiana is not contiguous, which induces non-planarity in the dual graph.

8.2 Number of Split Districts at each Census Level

The proportion of potential split districts in a state dual graph upper bounds the proportion of vertices to be removed to achieve planarity, i.e. giving the ϵ to be used in the ATAP definition. We examined nonplanar states at the county, tract, and block group levels, determining how many of the districts at each level in each state had irregular shapes. At the county level, the only state with a nonplanar dual is Louisiana, which has $1/64 \approx .015$ as an upper bound for ϵ. Our tract and block level results are documented in Tables 3 and 4. We see in general that each state only has a few basis points in proportion of potentially split districts at the tract level and even fewer at the block group level. The one exception is Alaska, but this might be skewed because of small amount of overall block groups (around 500 compared to others in the thousands) and the collection of small islands. We can use these results to find different approximations of ϵ, by taking the mean, minimum, or maximum of these individual estimates.

Table 3: Upper Bound ϵ by State at Tract Level

State	Upper Bound Epsilon
California	0.001 117
Colorado	0.008 807
Georgia	0.003 555
Louisiana	0.003 484
Massachusetts	0.002 029
New Jersey	0.000 995
North Carolina	0.001 366
Ohio	0.000 677
Pennsylvania	0.002 796
Virginia	0.004 195
Washington	0.009 602

Table 4: Upper Bound ϵ by State at Block Group Level

State	Upper Bound ϵ
Alaska	0.026 217
California	0.000 301
Colorado	0.003 397
Florida	0.000 349
Georgia	0.001 265
Louisiana	0.000 864
Massachusetts	0.001 422
New Jersey	0.000 316
New York	0.001 422
North Carolina	0.000 487
Pennsylvania	0.000 924
Rhode Island	0.004 907
South Carolina	0.004 576
Virginia	0.001 312
Washington	0.003 972

8.3 Non-parametric Graph Statistic Comparison

Table 5: DE Floodfill

Benchmark	Real Life Maps	Floodfill
Average Degree	-0.04554	-0.0552
Clustering Coefficient	0.3593	0.3576
Maximum Degree	15	9
Number of Leaves	0	0
Diameter	15	18
Radius	8	9
Prop. of Triangular Faces	.8889	.8456

Table 6: VT Floodfill

Benchmark	Real Life Maps	Floodfill
Average Degree	-0.0590	.0273
Clustering Coefficient	0.3711	0.3596
Maximum Degree	12	9
Number of Leaves	2	1
Diameter	15	14
Radius	8	8
Prop. of Triangular Faces	.9161	.8835

Table 7: WY Floodfill

Benchmark	Real Life Maps	Floodfill
Average Degree	0.0432	.0449
Clustering Coefficient	0.3912	0.3810
Maximum Degree	13	9
Number of Leaves	8	0
Diameter	10	12
Radius	5	87
Prop. of Triangular Faces	.9110	.8841

8.4 Real State Dual Graph Statistics

We plot real state dual graph statistics at the remaining Census levels below. Compute restrictions limited some of the statistics in the Census block dual graphs, given their size. Overall, we see the same broad characteristics as at the Census tract level, with a resounding majority of triangular faces and planar graphs, as well as the existence of certain tail outcomes like states with high numbers of leaves at different levels.

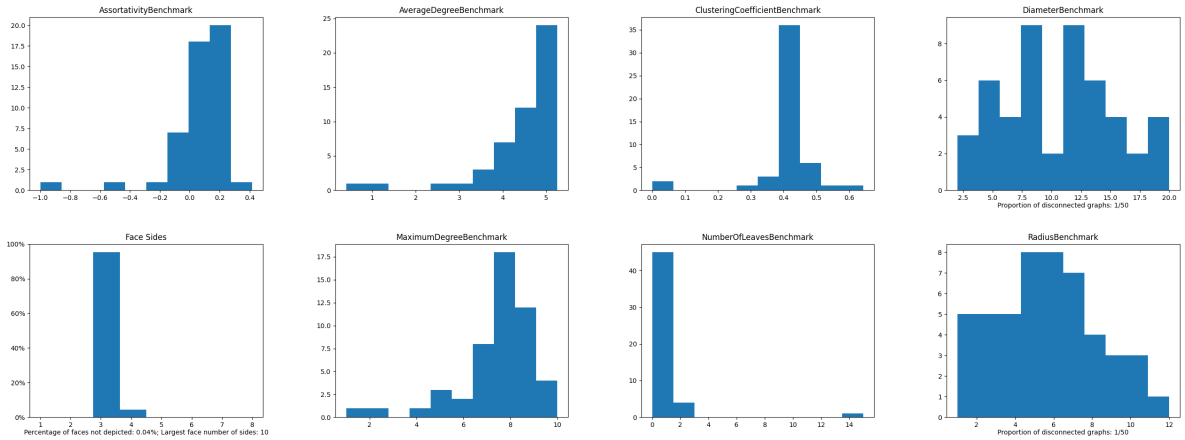

Figure 6: Real County State Dual Graph Statistics

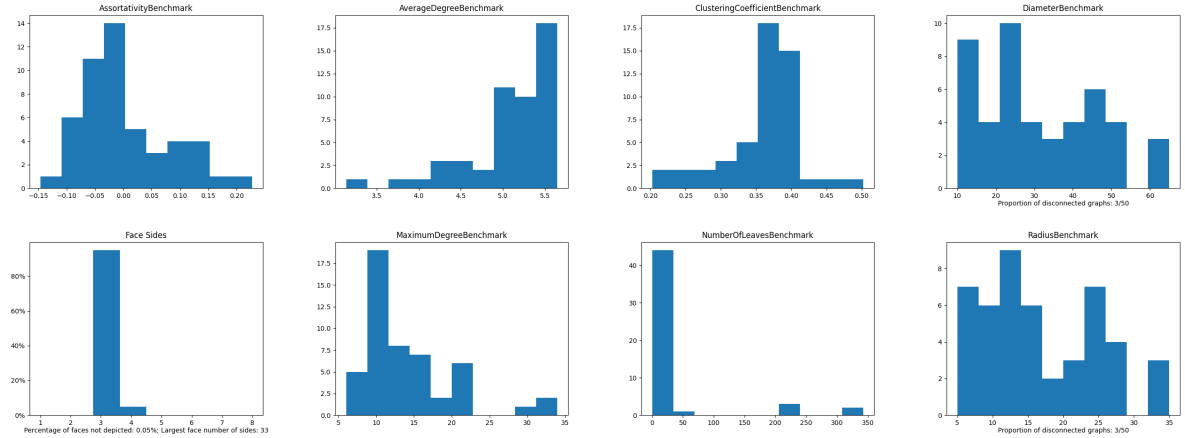

Figure 7: Real Census County Subunit State Dual Graph Statistics

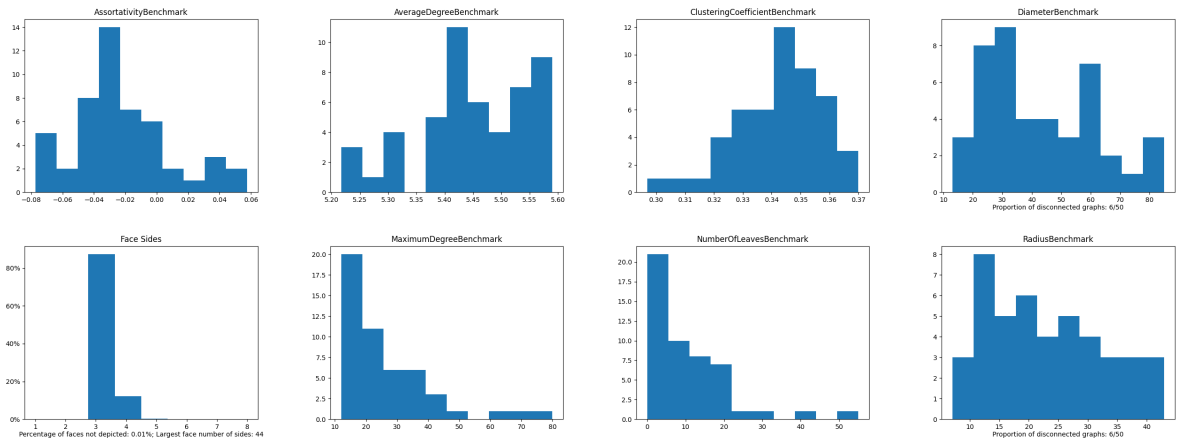

Figure 8: Real Census Block Group State Dual Graph Statistics

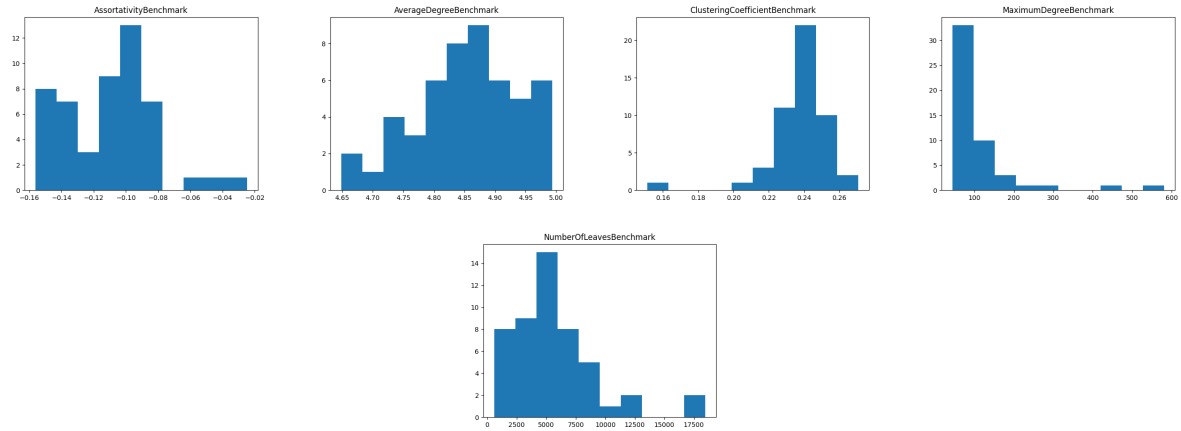

Figure 9: Real Census Block State Dual Graph Statistics

8.5 Tail Outcomes on Dual Graphs

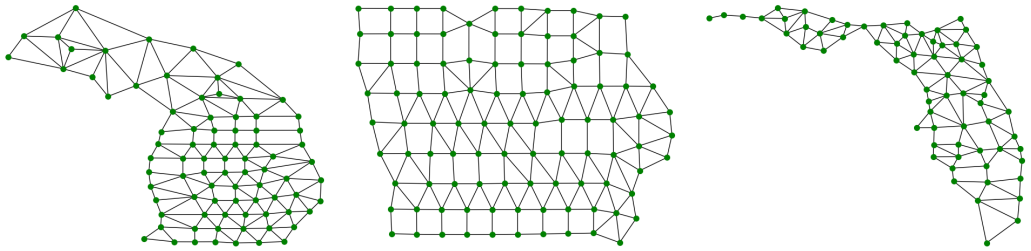

Figure 10: First, a dual graph of Mississippi on the county level, which is both planar and almost-triangular. Then, two “edge" outcomes of county duals we would still like our algorithms to be able to generate: the Iowa and Florida duals. Notice that while Mississippi and Iowa have a similar number of counties (82 vs 99, respectively), Iowa’s graph is far less triangular—this is an artifact of the grid-like division dating back to the Public Land Survey System established under the Land Ordinance Act of 1785. Similarly, because of Florida’s panhandle, there is a much heavier tail of low-degree nodes in its county dual.

8.6 Planarity and Face Distributions, By State and Level

To understand just how overwhelmingly triangular dual graphs for states are, we visualize the distribution of faces state-by-state for states with planar graphs at all Census levels (except block for computational reasons), as opposed to combining all of the faces from all states as in 8.4. We note that 50-100% of the duals are planar at every level, with 49/50 states being planar at the county level, 26/50 at the county subunit level, 39/50 at the tract level, and 34/50 at the block group level.

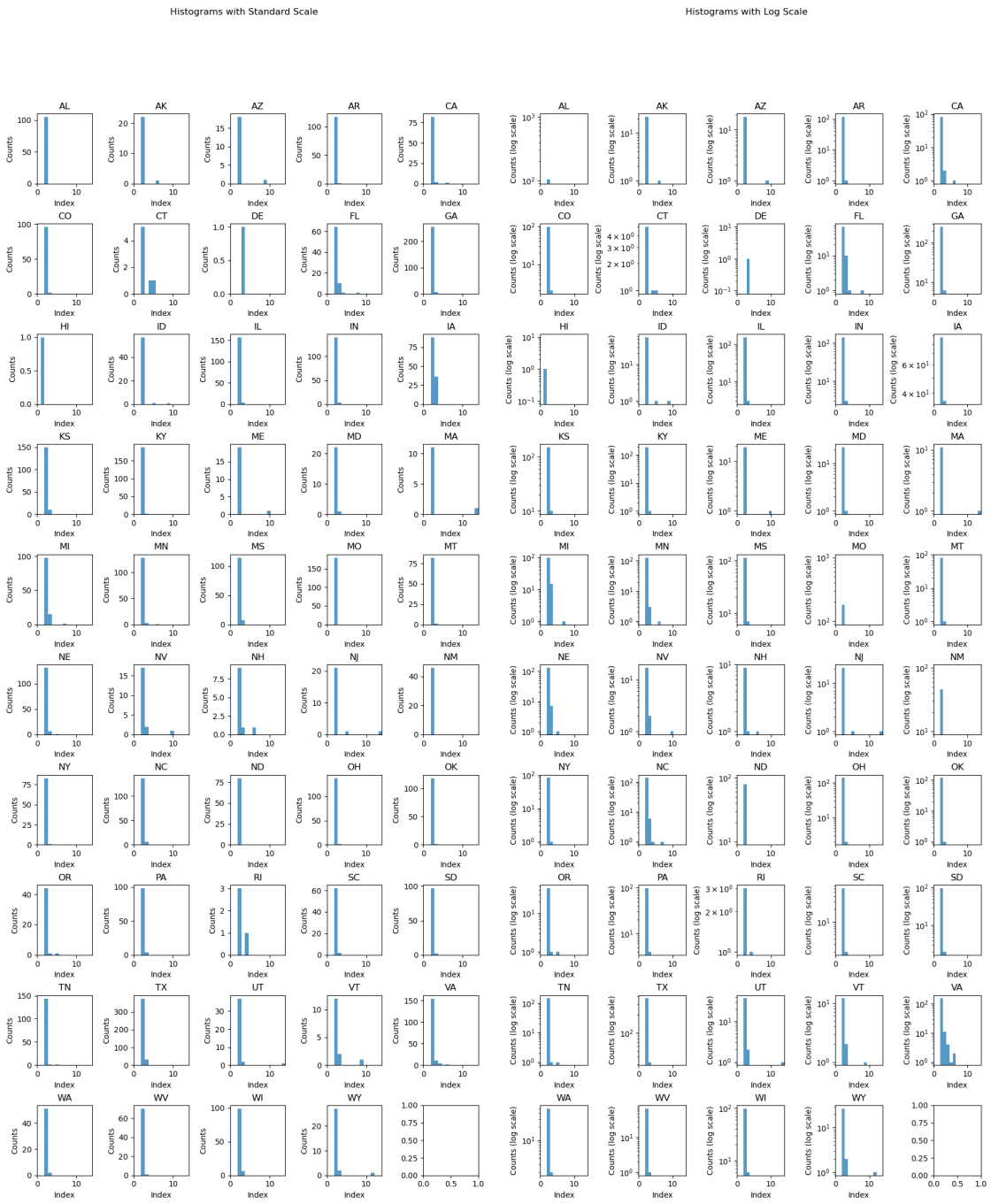

Figure 11: Planar County Dual Face Histograms (Non-Log and Log)

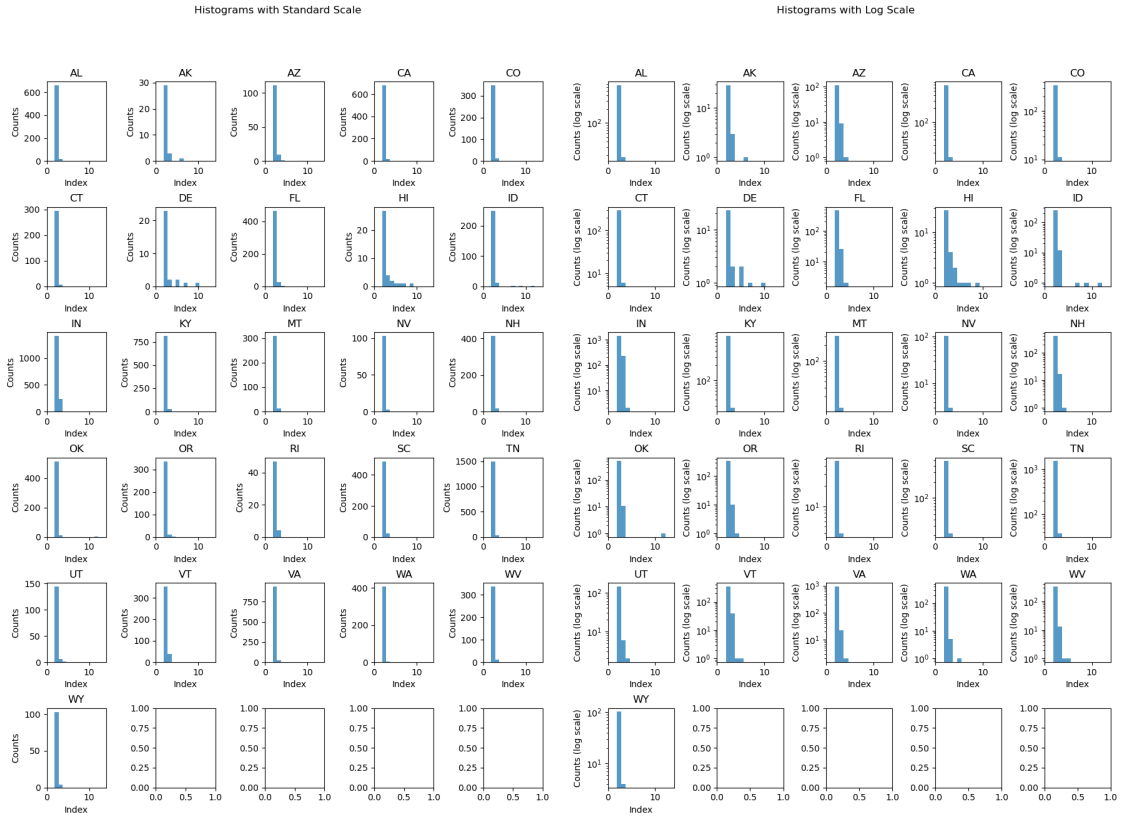

Figure 12: Planar County Subunit Dual Face Histograms (Non-Log and Log)

Figure 13: Planar Tract Subunit Dual Face Histograms (Non-Log and Log)

Figure 14: Planar Block Group Subunit Dual Face Histograms (Non-Log and Log)