**Figure 1: Random Potential**

potential realization in real space (several versions)

V1,V2,V3 different realizations

V1.1,1.2,1.3 : Updated labels! Newest.

**Figure 2: Density of States and Related**

a: Overall DOS for System Sizes 64, 128, 256, 512, 1024, 2048

b: 1024 DOS Points + FWHMs + inner window

Put FWHM infos here.

c: Non-zero Chern scaling plot:

8,16 system sizes excluded from fitting

Fitted parameters: $A = 0.2595 ± 0.0003$, $b = 2.4967 ± 0.0030$

R-squared: 0.9999

RMSE: 0.2554

**Figure 3: Nearest Neighbor Level Spacings (tail)**

First, let’s define some notation:

and then, given a specific set of (for example, within the tail windows)

we may define  where is the average for that dataset.

Eigenvalues in |E| ∈ [3.79, 5.75]: 45065 / 21731328 = 0.21%

99.5% of data is shown in histogram 0.5% tail is cutoff (messy values)

The exponential line is **not a fit**, I simply compute the KS-statistic and chi-squared *assuming* this distribution

\*Can get higher p-values with smaller window, but this window has reduced chi-squared nearest to 1, indicating a very great fit! This p-value >5% already indicates no statistical difference between distributions so its sufficient.

**\*\*Note: for the sake of the fitting metrics (KS-Statistic, Chi-squared), the data for P(s) and P(r) plots is not symmetrized: the histograms are identical with/without symmetrizing, and symmetrizing has the result of worsening the fit metrics since there’s essentially 2x every real separation value.**

**Figure 3.5: Nearest Neighbor Level Spacings (center window)**

Different Chern scenarios: Within each trial, the eigenvalues are cut-off within the specified energy window. Then, they are filtered to match the specific Chern scenario.

Linear Plots: All Chern- All data. C=0: 99.45541565092927% of data. C=1: All Data

All other plots: all data shown

Fitting: say that . Then, our normalization coefficients are given by

Then, we may fit the functional form using negative-log-likelihood (least squares) optimization, either fixing certain parameters or letting them to be free. This minimization is done with linear scaling, ie, the residuals in linear space and not log-transformed residuals.

**Figure 4: Second Order Nearest Neighbor Level Spacings**

Same as previous figure, now with 2nd order spacings.

and then, given a specific set of (for example, within the tail windows)

we may define  where is the average for that dataset.

For simplicity, let’s define to reduce notational complexity.

Figure 6: (UNFOLDED) Nearest Neighbor Level Spacings

**Figure 7: First-Order r ( ) vs. E for N=1024**

Our notation is that

* Includes three panel and single-column plot for each Chern case
* GUE Expected Value:
* Poisson Expected Value:

Each is plotted at its respective value.

How is binning accomplished for the averages?

Bins represent QUANTILES of the data! So, the ’s (and the correspond ) are first sorted and then split up into small groups to average the values. The first and last datapoint are 0.25% of the data, respectively. Then, there are 35 central bins which represent 99.5% of the data, or, of the data each.

**Figure 8: First-Order P(r) & Fit (inside [-0.03,0.03])**

First, for we define . Then for , we can complete a change of variables and define a new probability distribution such that:

where is the original function and

( there is no analytical form, especially since may be non-integer valued )

* Pearson chi-squared test: where
  + are expected bin counts, are observed : use Freedman Diaconis Estimator to get ideal bin counts for consistency
  + Num FD Bins – params – 1 “(k-m-1)”
* KS-stat and corresponding p-value

\* Each is associated with the given . However, we only constrain so or may lie outside the given window

**Figure 9: Second-Order r () vs. E**

so

* GUE Expected Value:
* Poisson Expected Value:
  + <https://doi.org/10.1103/dkgc-4bd1> has formula for this non-overlapping case, originally from supplemental section of <https://doi.org/10.1103/PhysRevResearch.2.032063>
  + Poisson: so then on the folded regime, the average is precisely .

**Figure 10: Second-Order P(r) & Fit (inside [-0.03,0.03])**

* Same methods as Figure 8 described above, this time with the non-overlapping ratio
* We fit the same functional form to get a new “beta”
* Same , though we expect
  + with for j=2
  + So, if we’re fitting , we can retrieve
    - Clearly, besides for all Chern, this relationship does not remotely hold between the P(r1) and P(r2) expected beta parameters.