Arbitrary TN Approximate Contraction

Overview of Steps

1. Find a ***contraction tree***

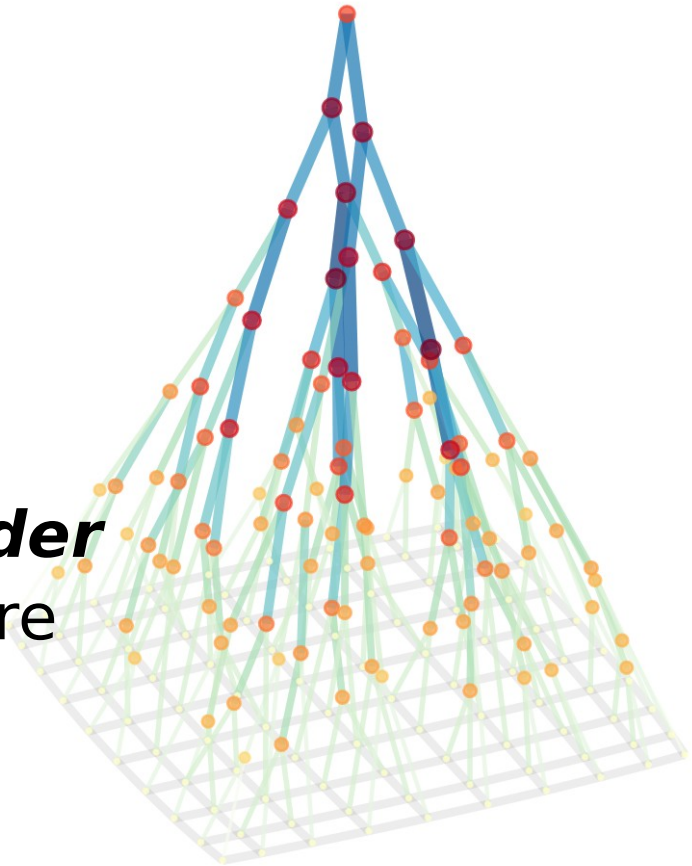
2. Find a contraction tree ***ordering***

3. *As we proceed up tree:*
    1. Find ***multibonds*** to compress
    2. ***Gauge*** around multibond tensors
    3. ***Compress*** between multiband tensors

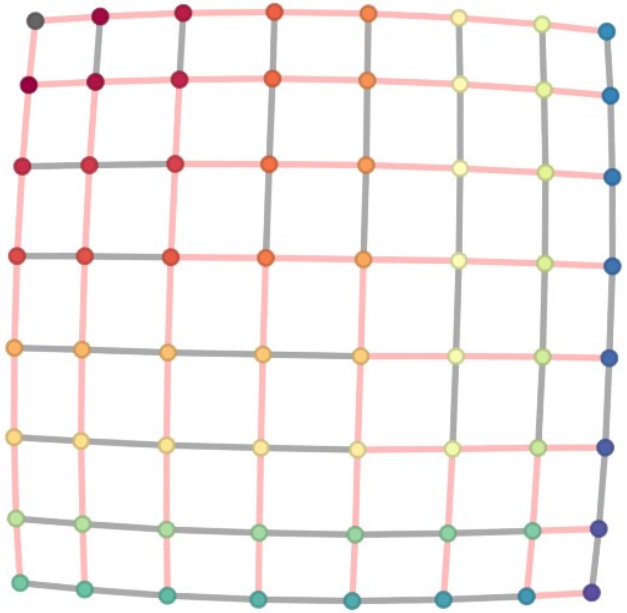
Choose Contraction Tree

(all spanning trees map to contraction trees but not other way round)

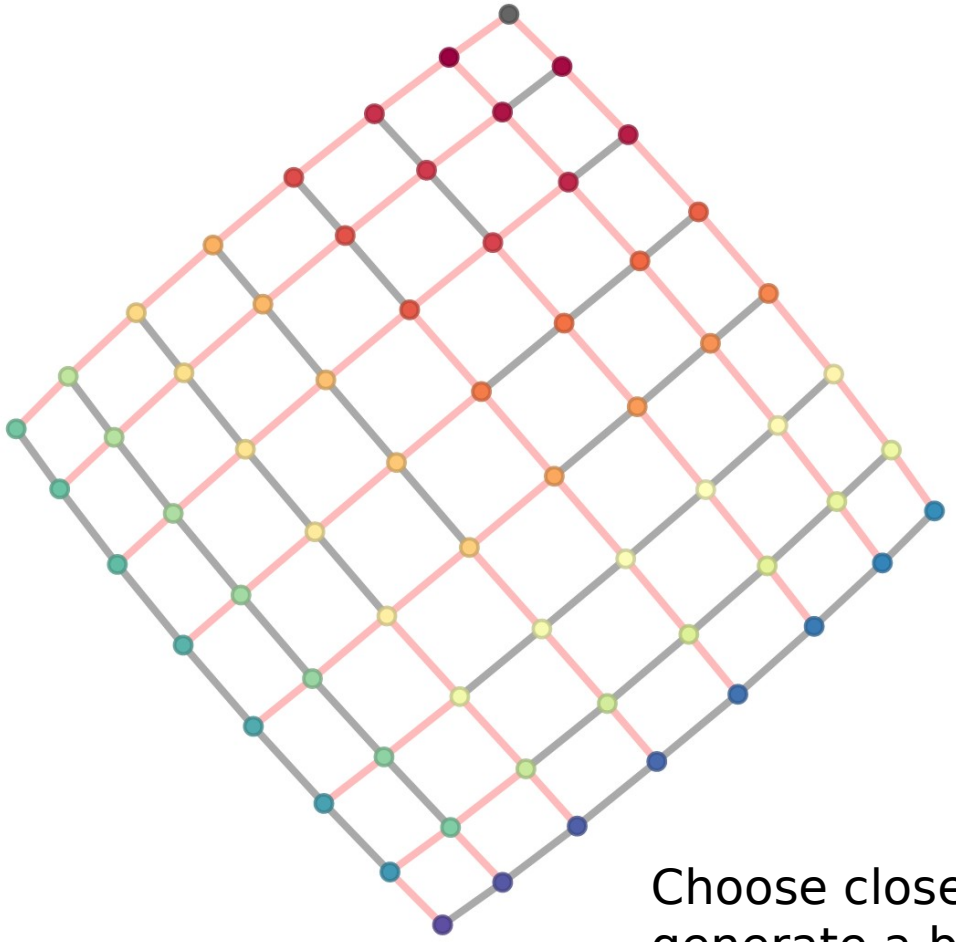
- Generate as usual (with tweaked heuristics):
  - Need to carefully choose ***order***
  - Maybe more efficient?



- Generate from a spanning tree of original graph
  - Easy to visualize ‘surface’
  - *Easier* to define order
  - Can contract ‘around’ stuff

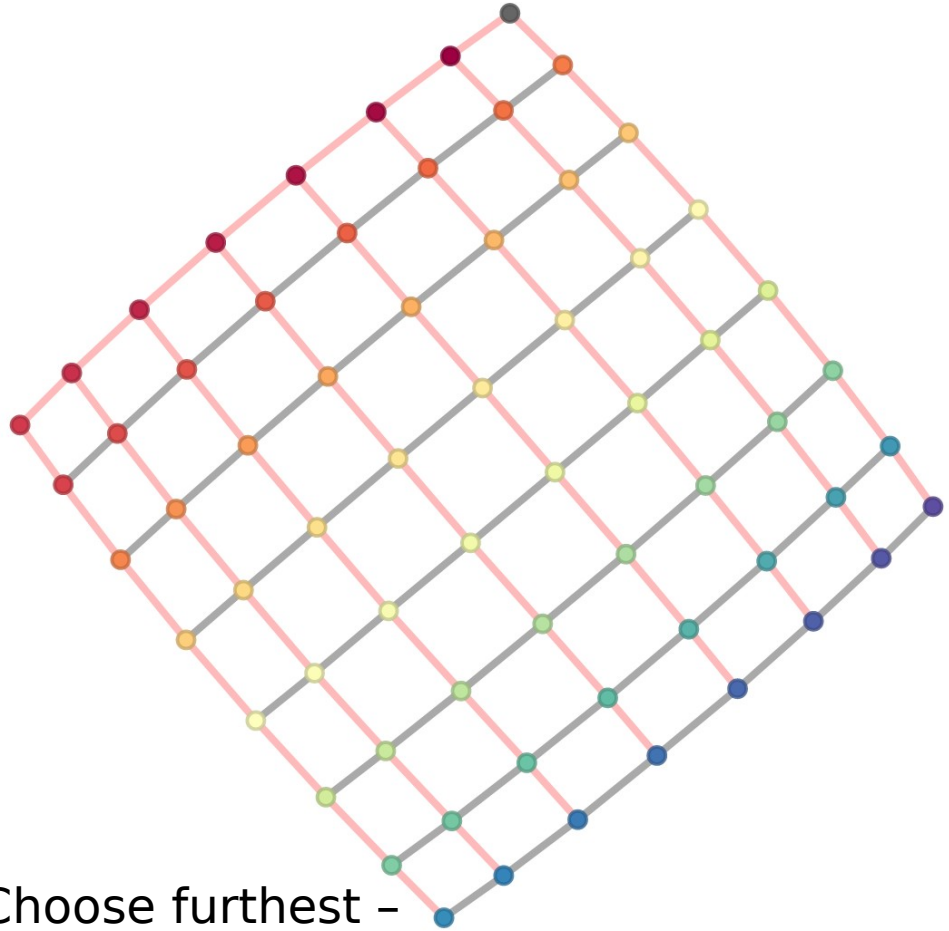


Spanning Tree Generation



Choose closest – generate a ball

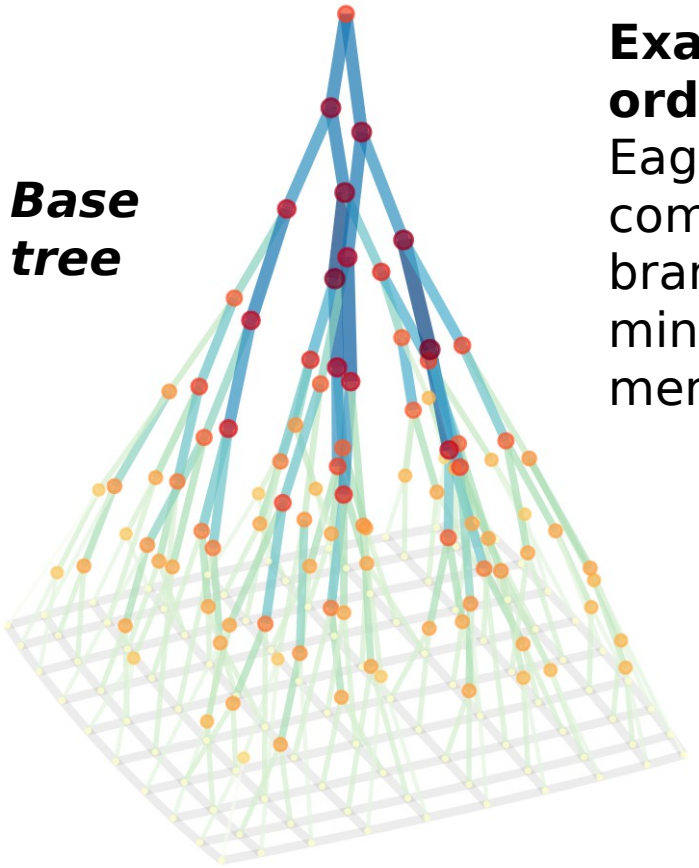
1. Start with particular region
2. Expand region by adding tensor most connected (minimizes ‘surface tension’)
3. When all equally connected have a choice …



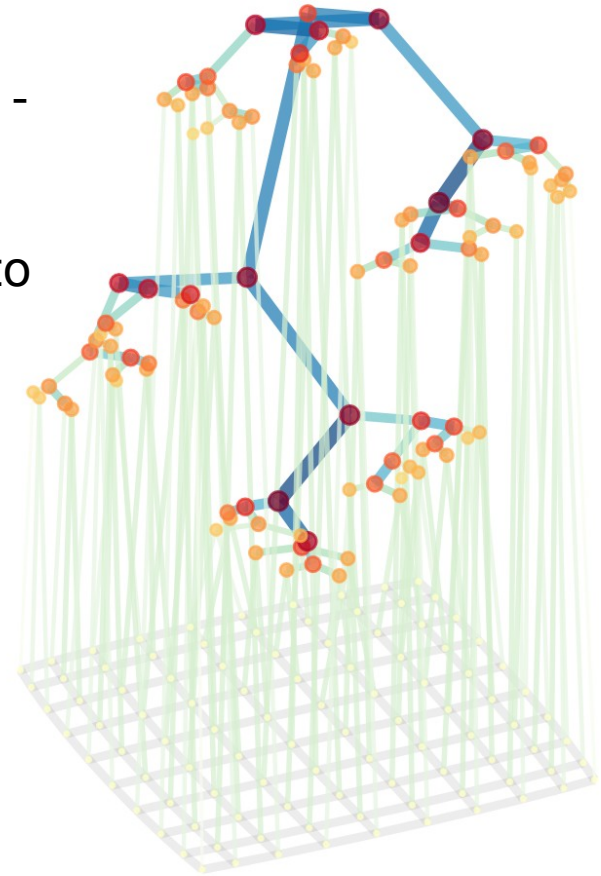
Choose furthest – generate planes

Contraction Tree Ordering

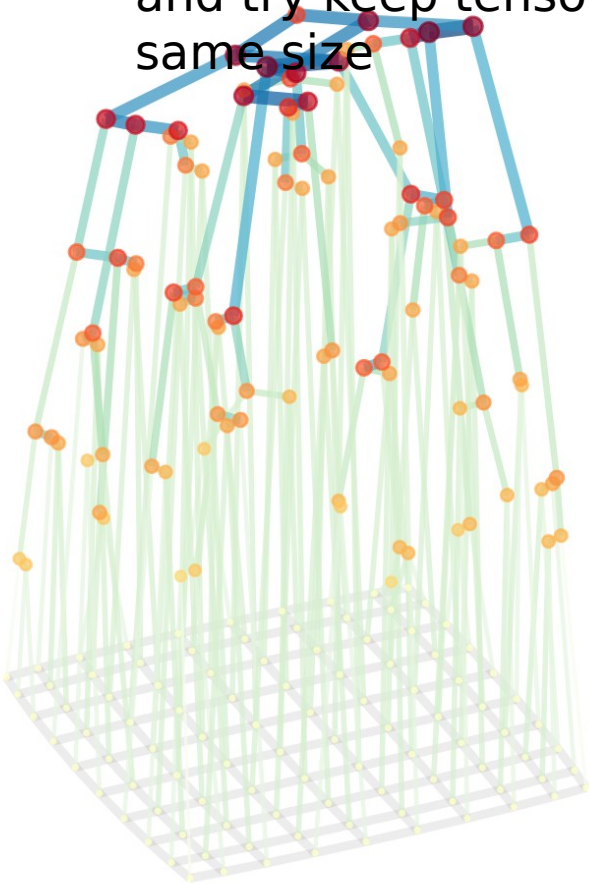
Base tree



Exact ordering - Eagerly complete branches to minimize memory



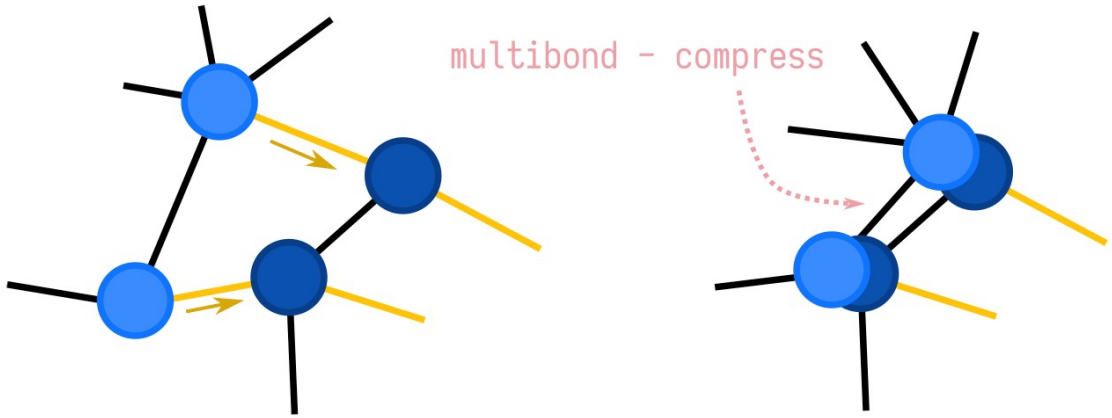
‘Surface’ ordering - Contract ‘inwards’ and try keep tensors same size



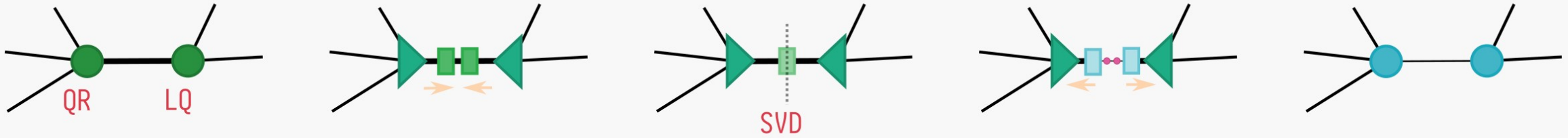
Compress as we contract

As we generate tensors with multiple bonds to other tensors – compress them

Probably want some form of arbitrary geometry gauging first



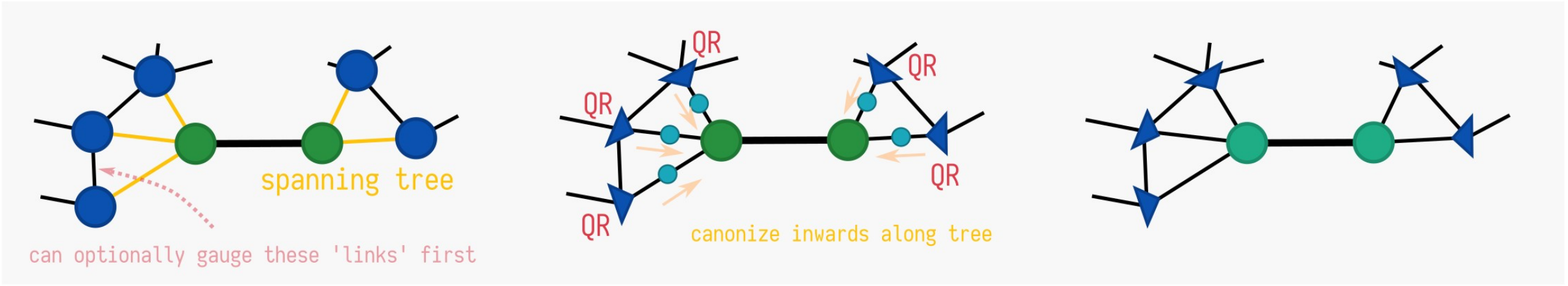
basic reduced compress



Local Canonization

- Just absorb QR factors inwards from neighbours
- For trees is exact canonicalization

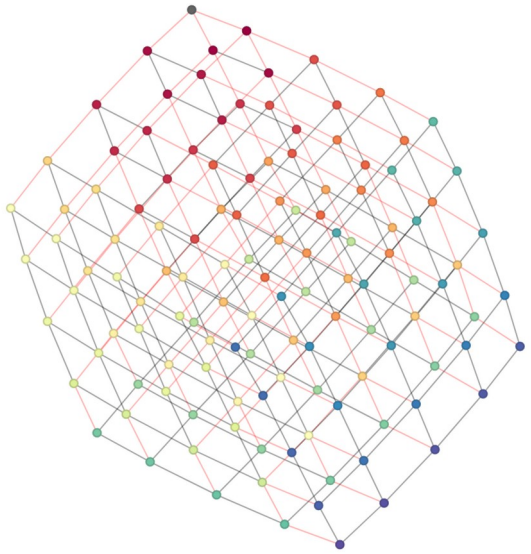
local canonization "perfect for trees"

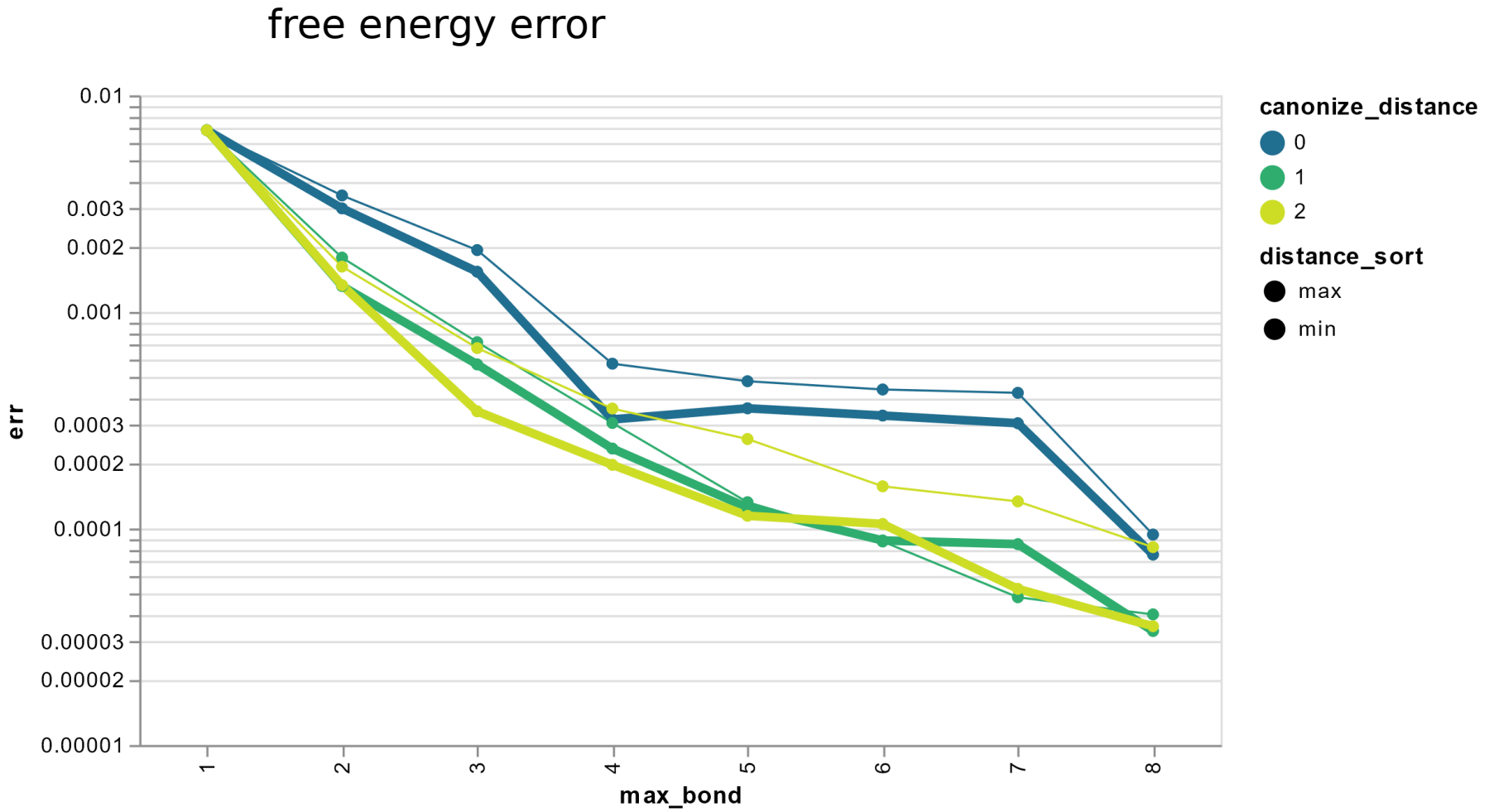


Spanning Tree Results (boundary-style)

6x6x6 OBC
Classical Ising Model

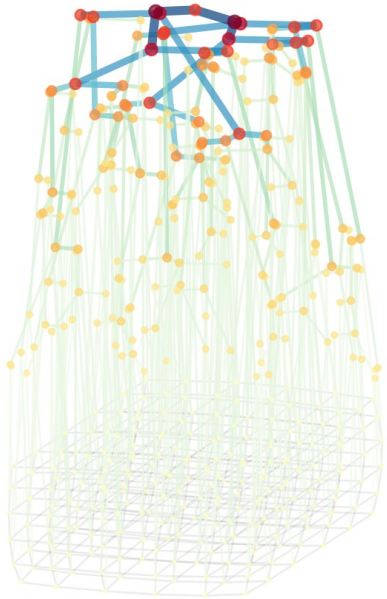
(close to critical)

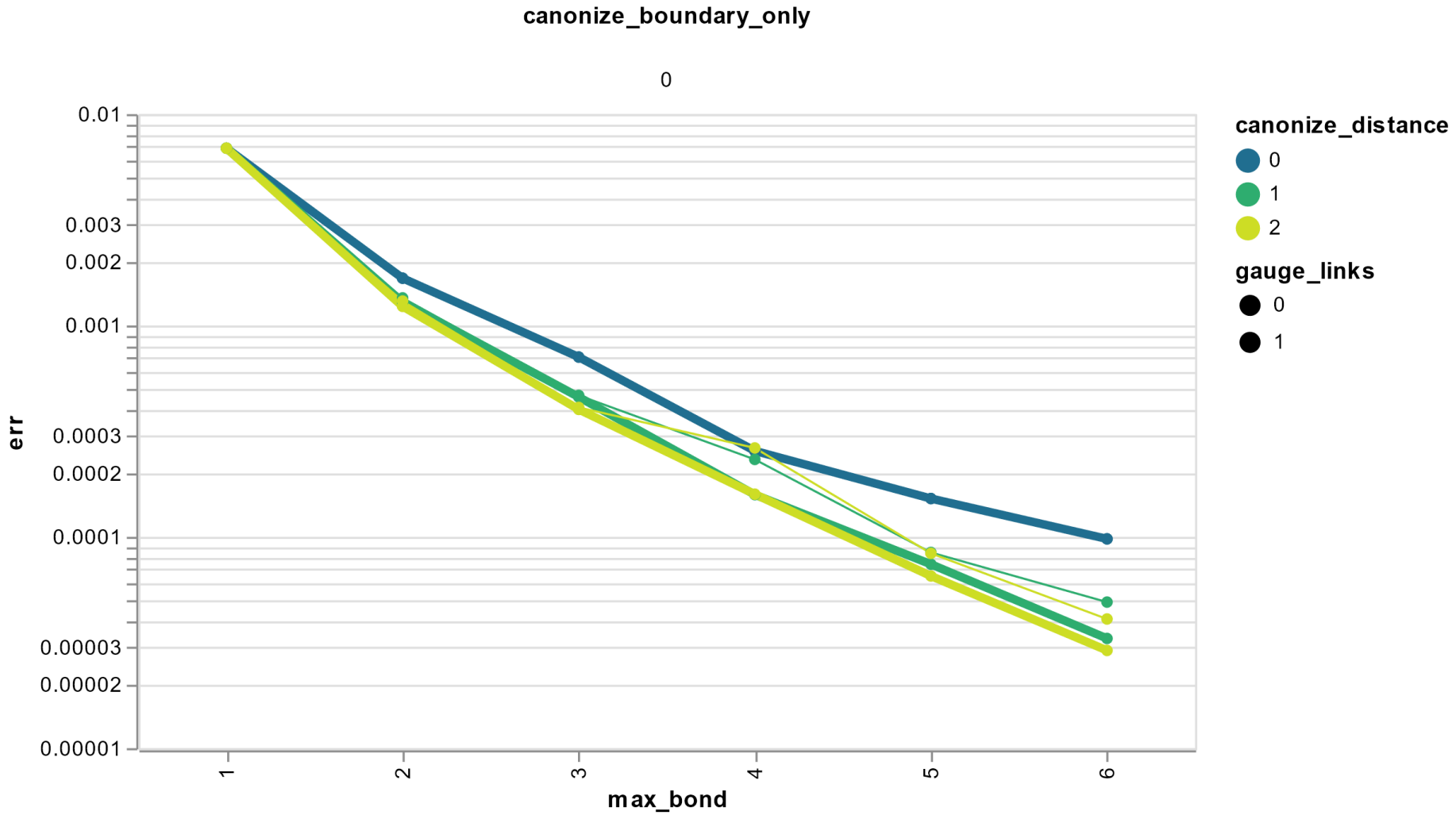




Balanced Tree Results (‘RG’-style)

Very similar scaling or results at slightly lower chi (though memory scales higher?)

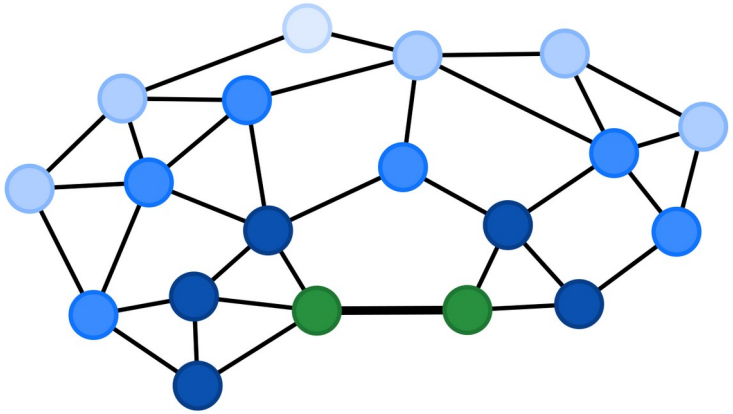




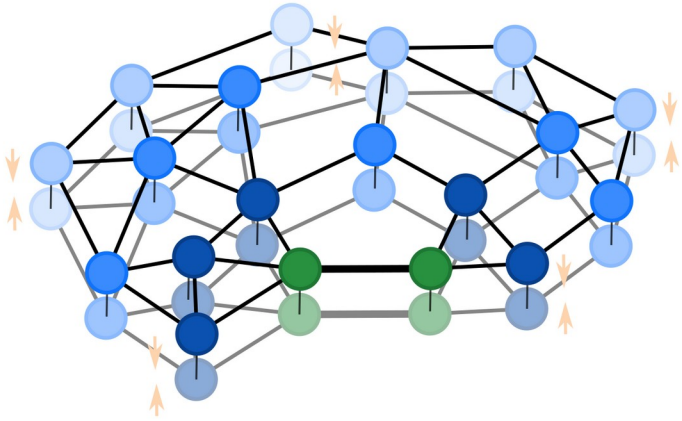
Flavours of Arbitrary Gauging

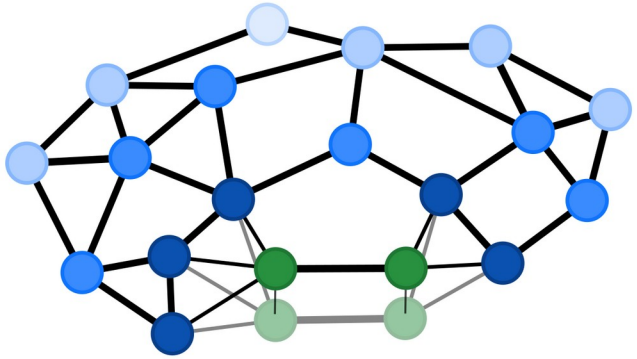
- If we want to gauge a bond, options depend on type of TN

scalar TN



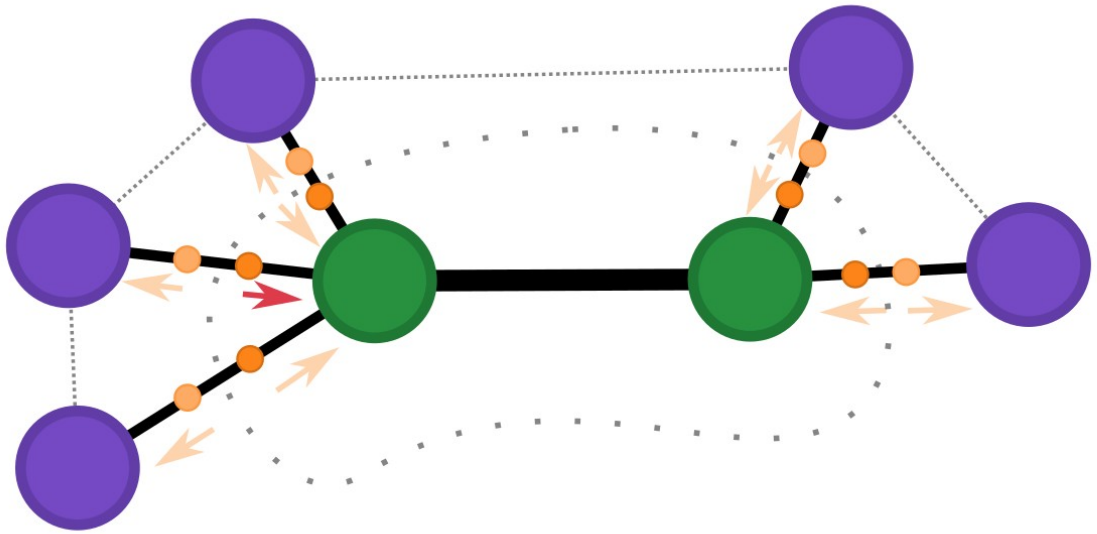
vector TN

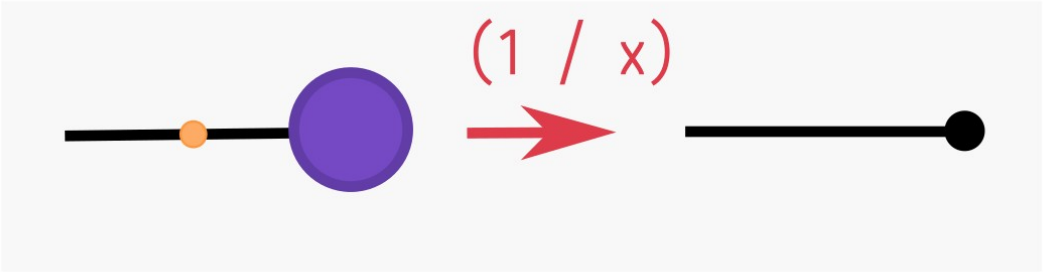




And also whether we want to guage **‘permanently’** or **‘temporarily’**

scalar TN product gauge

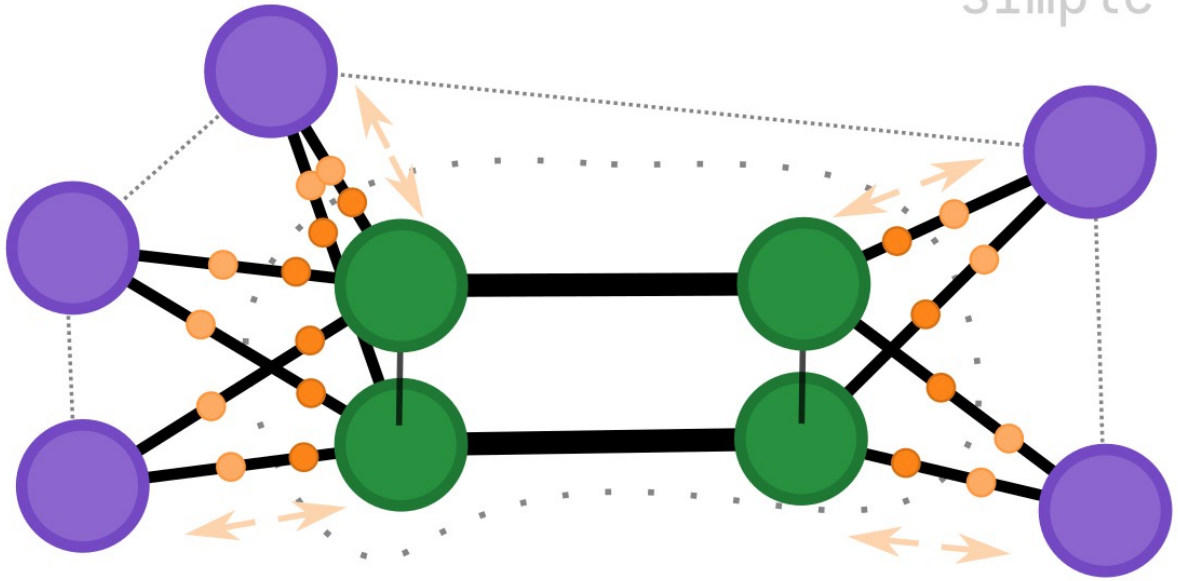


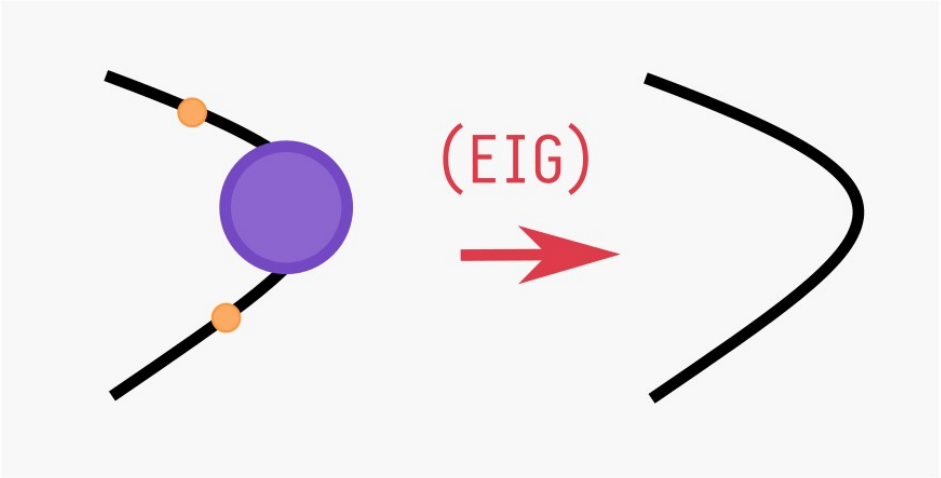


Since env is product, we can insert gauge back into original TN bonds

vector TN product gauge

"simple update is not the only choice"



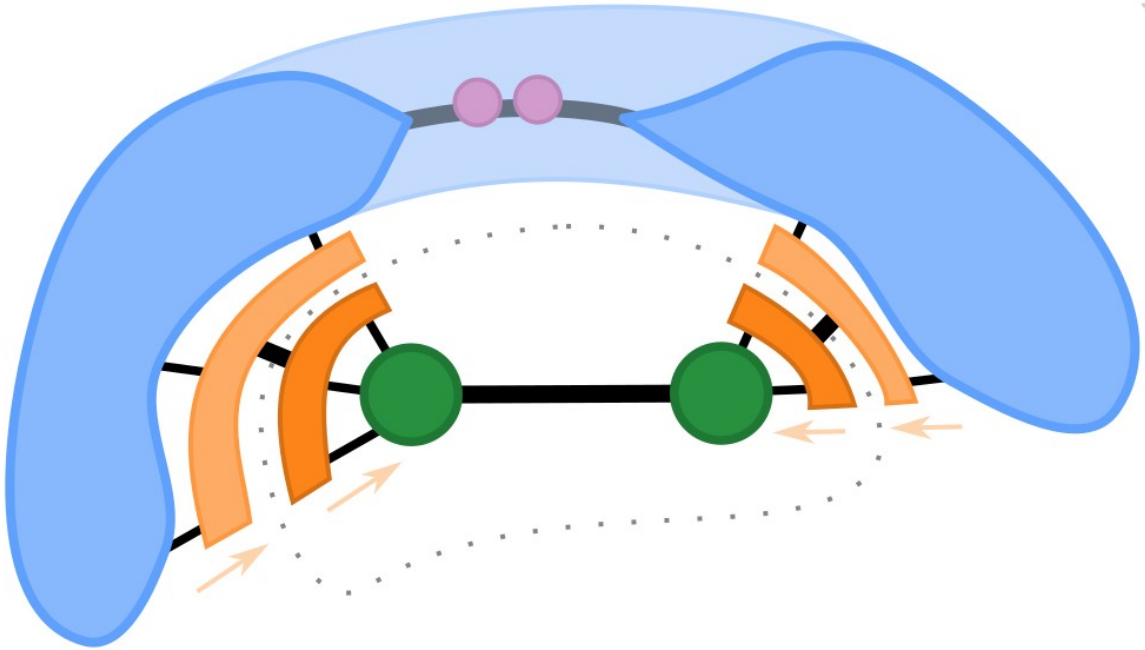


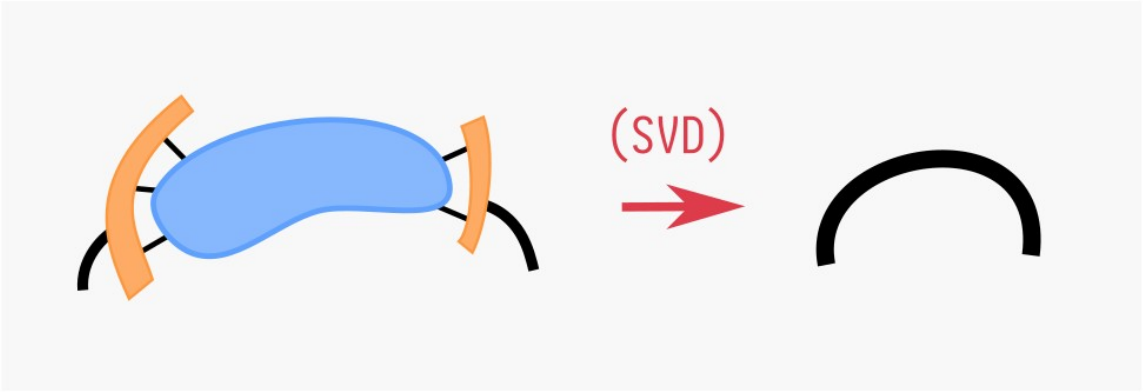
Product env between bonds means we can still gauge original TN

Simple Update (= Belief Propagation) is *a* (diagonal) choice of this, but not necessarily the best.

scalar TN full gauge (temporary)

"second renormalization"

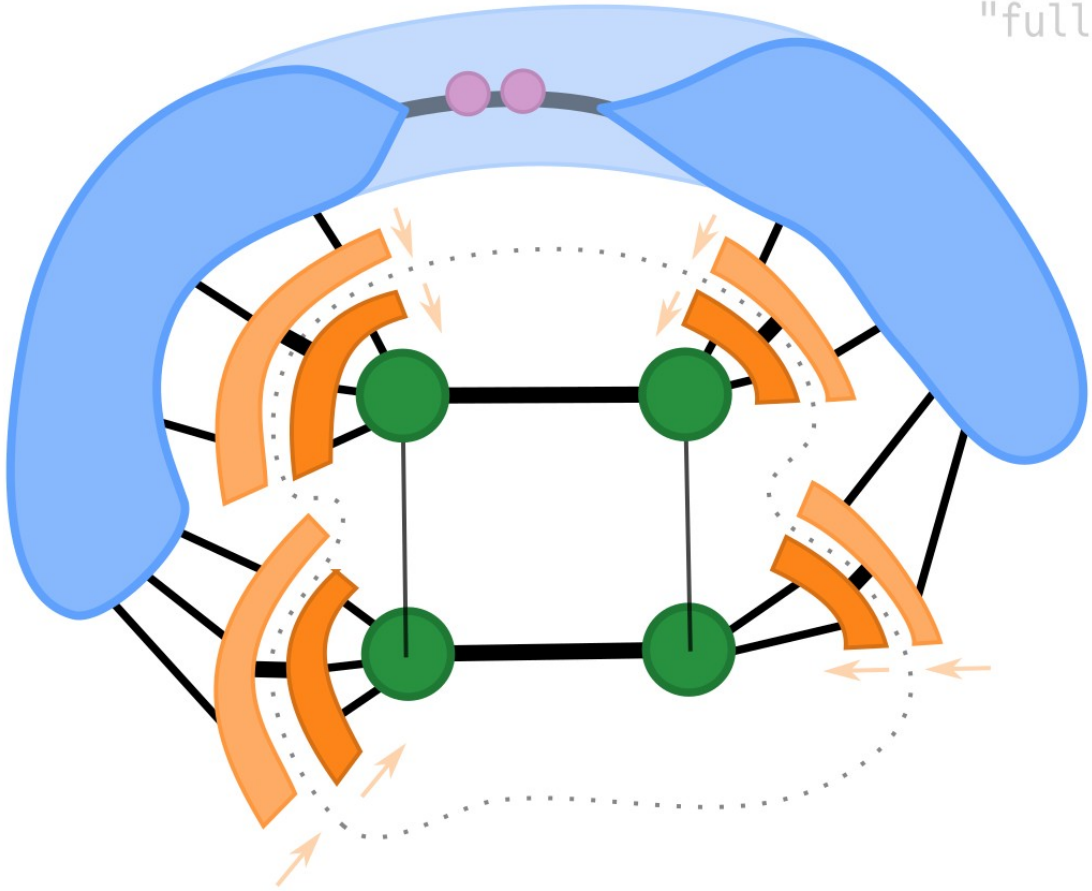


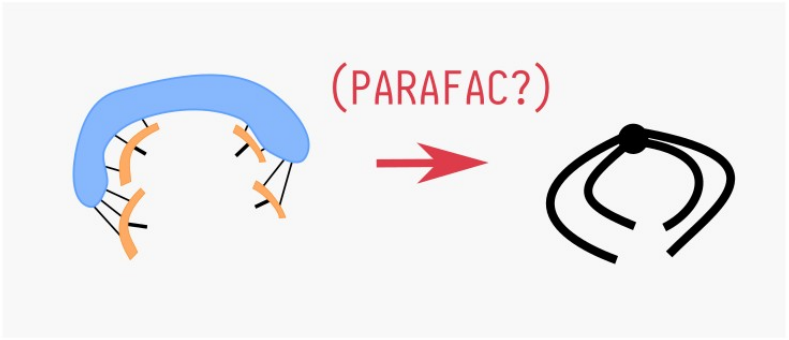


Take into account correlations between outer bonds – means we can only ***temporarily*** gauge tensors

vector TN full gauge (temporary)

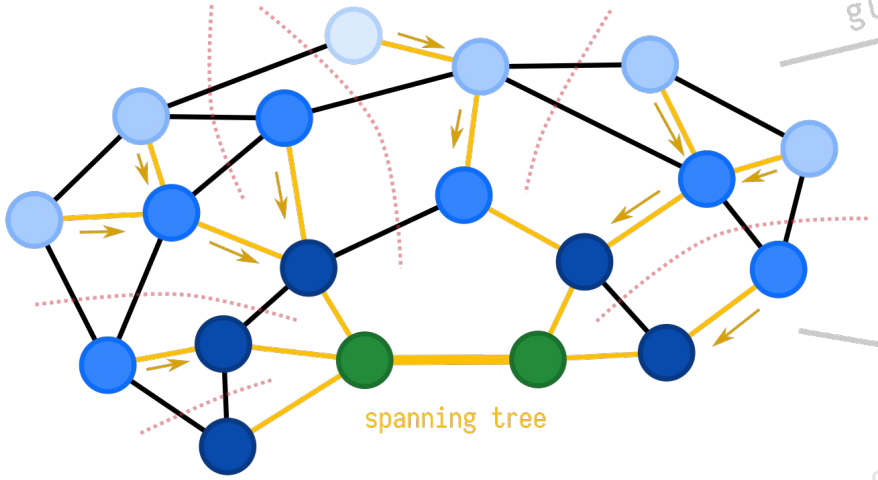
"full update without the fitting"

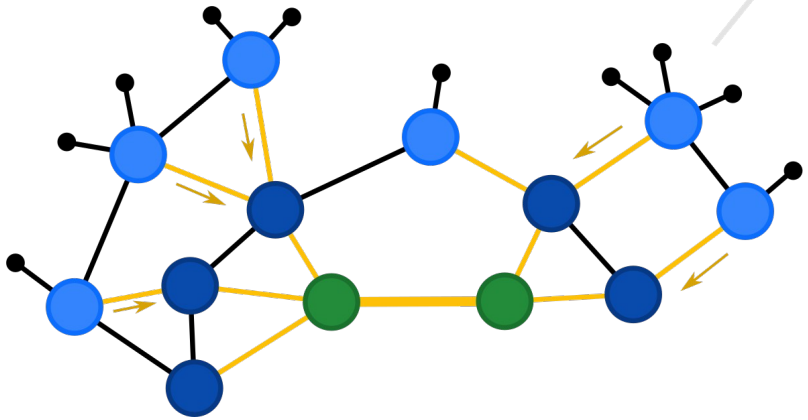


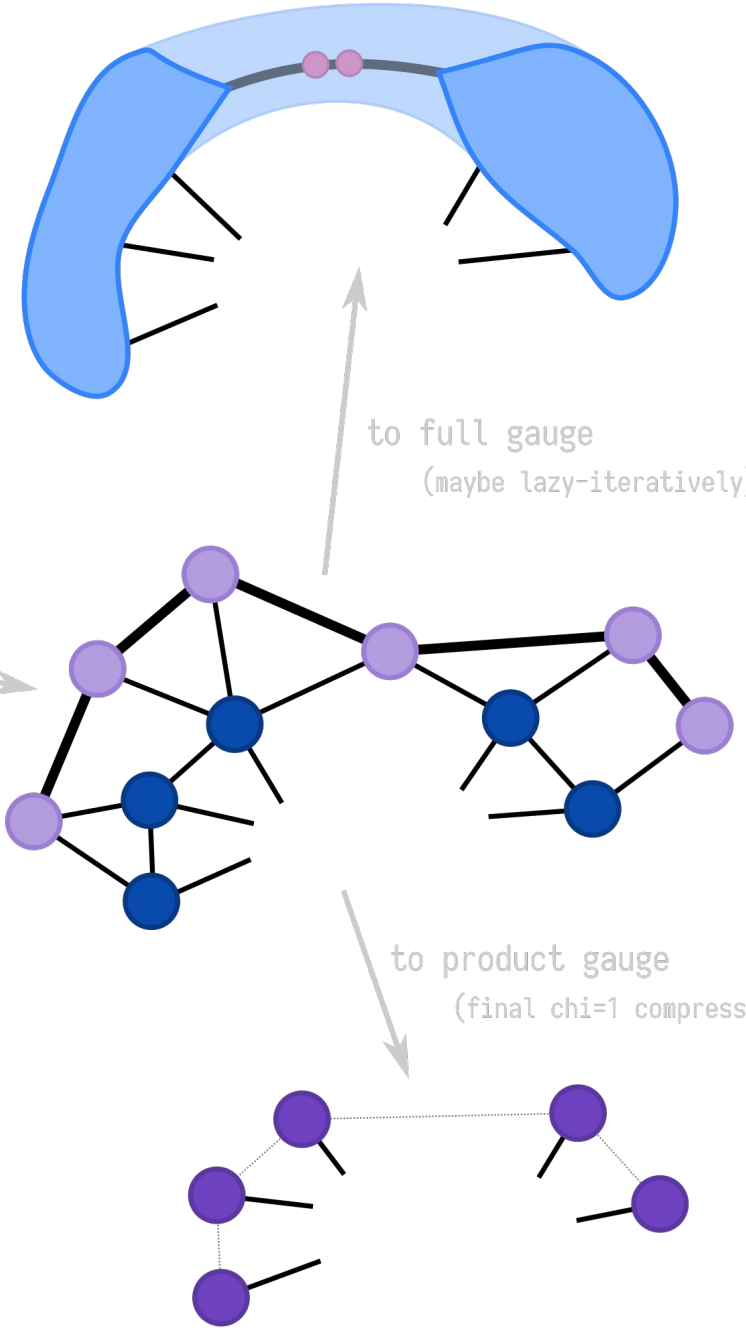


4-leg structure means we can’t use SVD, might be a more robust way of doing Full Update?

how do we generate the environment?







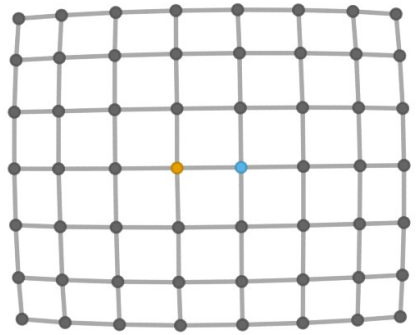
Main gauging options

- LC
- LPG
- GPG
- GFG

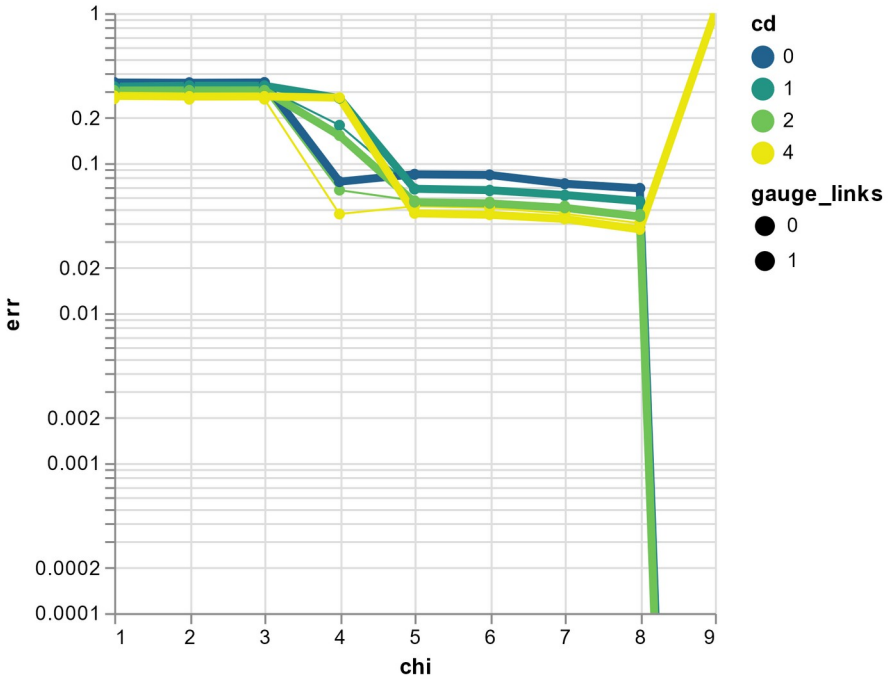
Results for single bond compression

(scalar TN gauging)

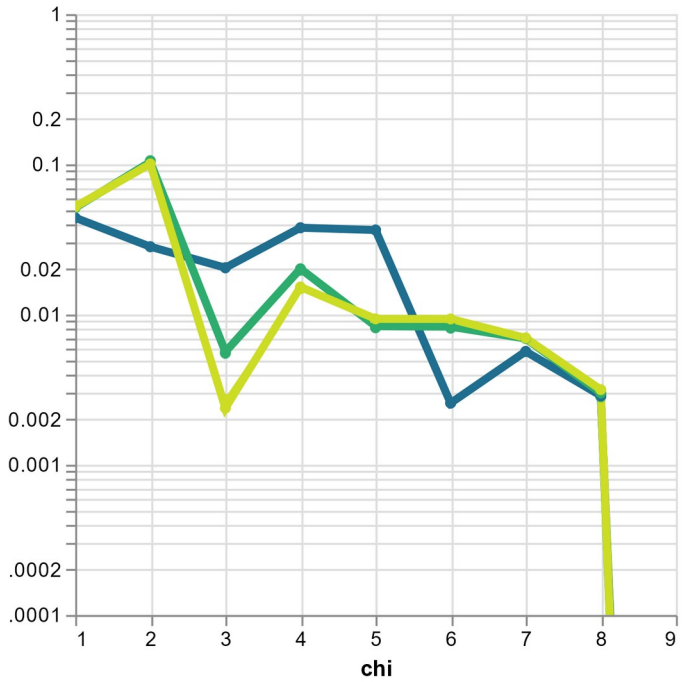
8x7 D=3 PEPS norm contraction



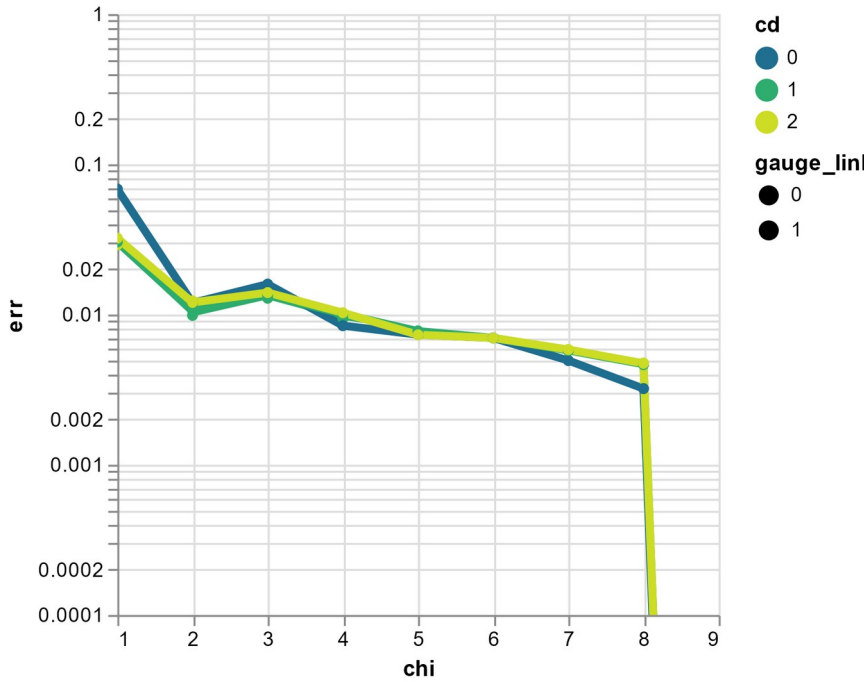
LC



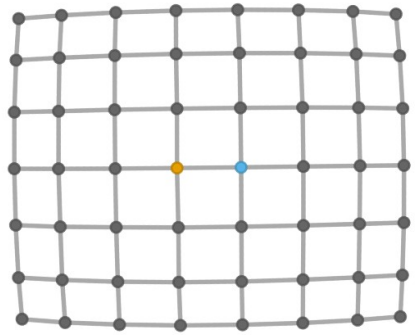
LPG



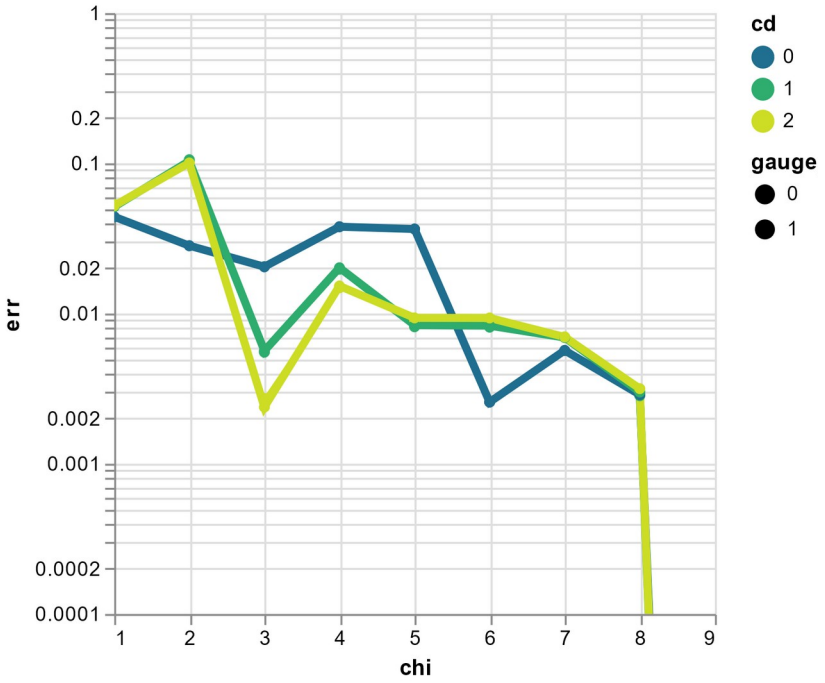
GPG



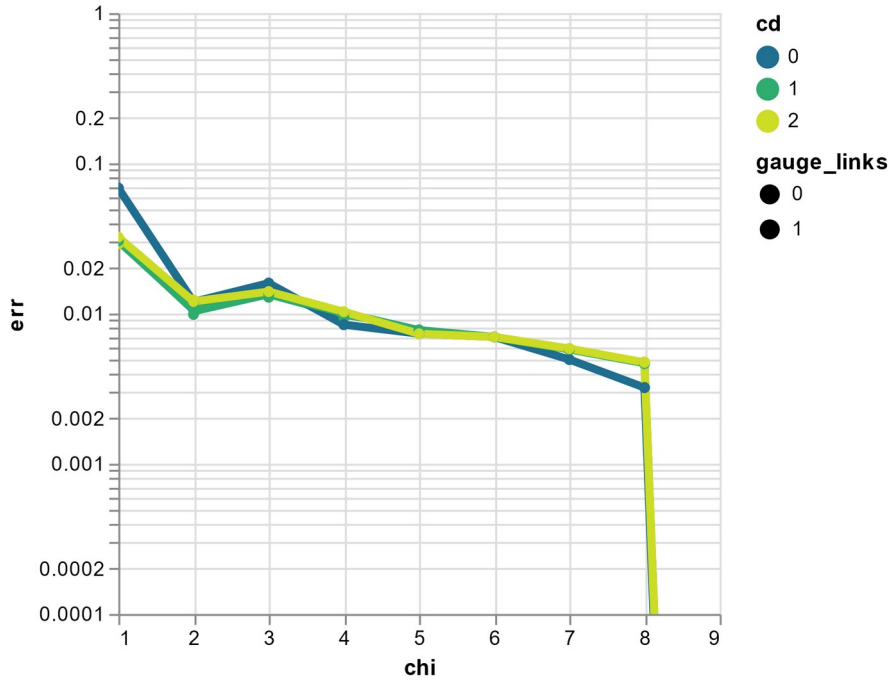
Results for single bond compression



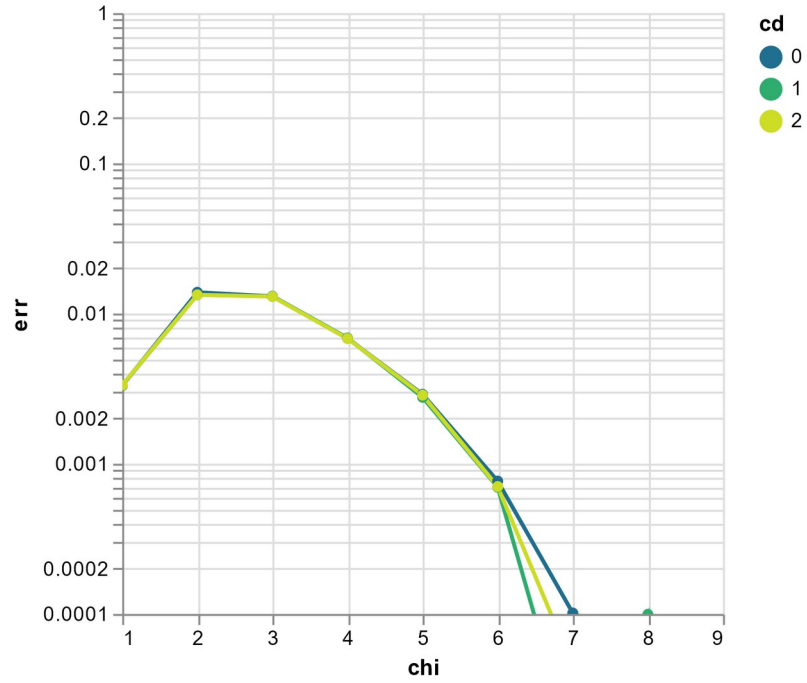
LPG



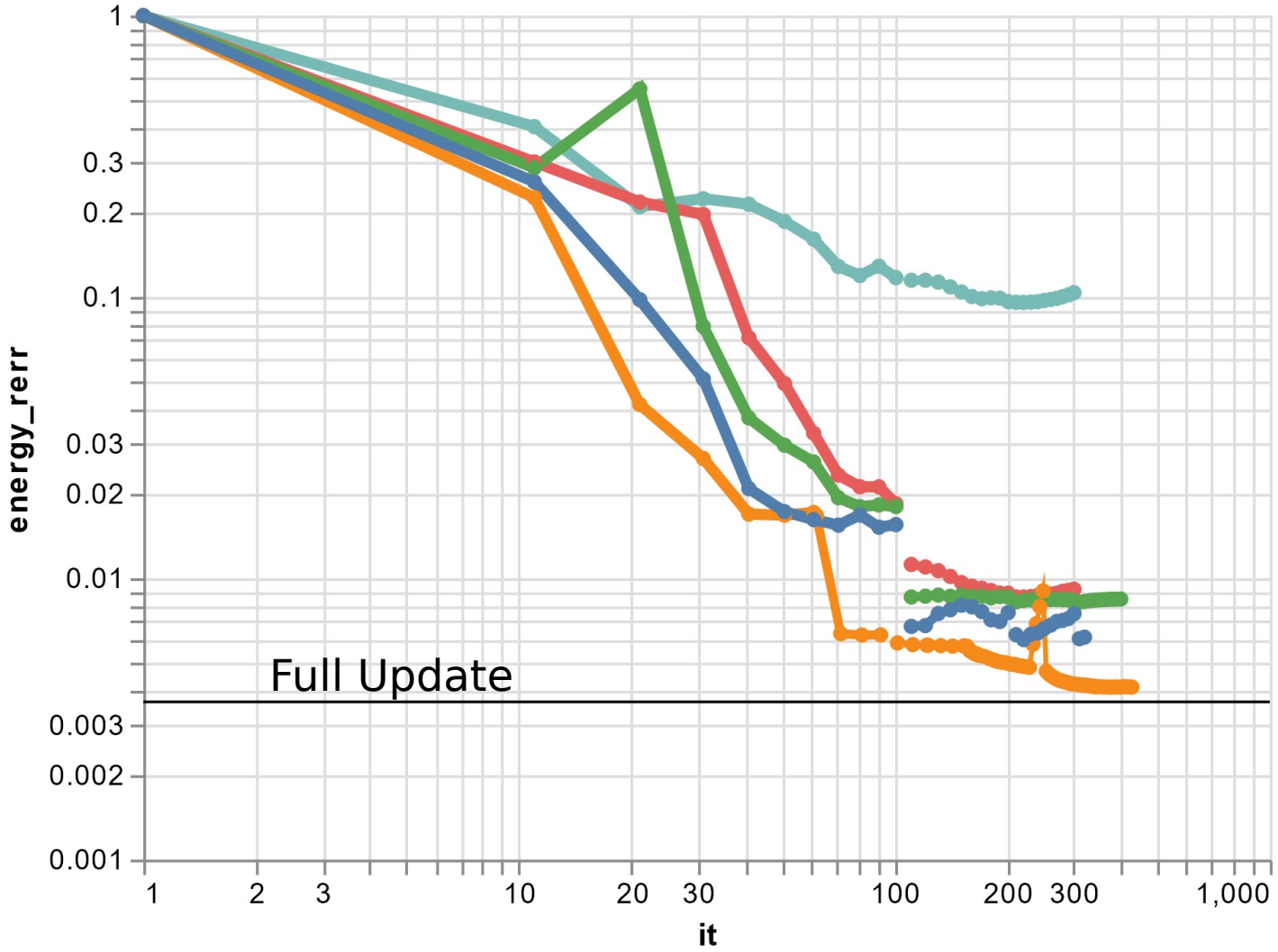
GPG



GFG



Real example - TEBD 2D Gauging



(vector TN gauging)

10x10 D=4 Heisenberg Model

- LC similar to SU
- LPG much closer to FU than SU

Summary

- Can handle tree or span – generated contraction sequences
- Results seems promising for 3D classical Ising model (easy TN)
- Might be able to use much better local gauging + full gauging
  - Best way of cutting dangling bonds?