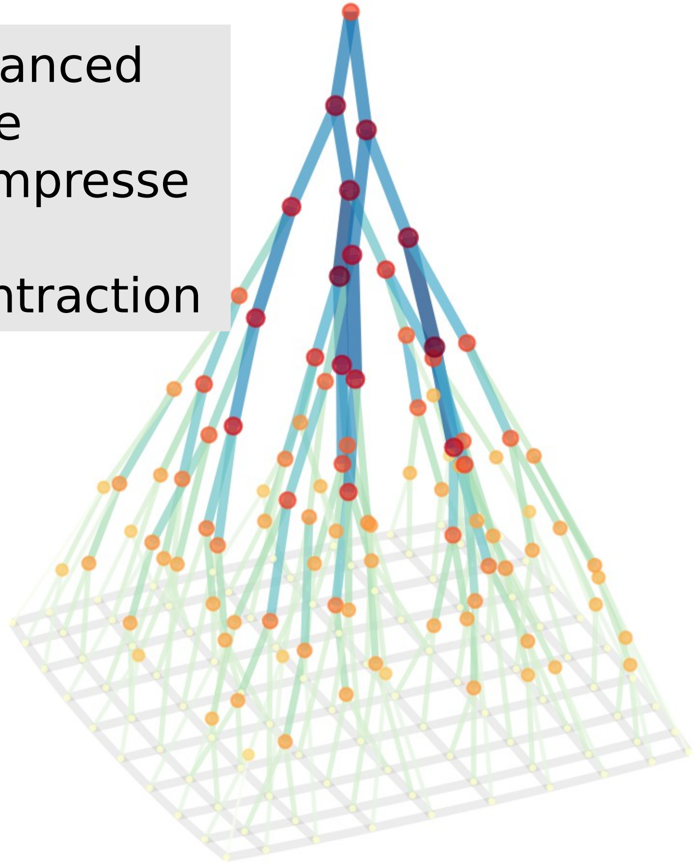
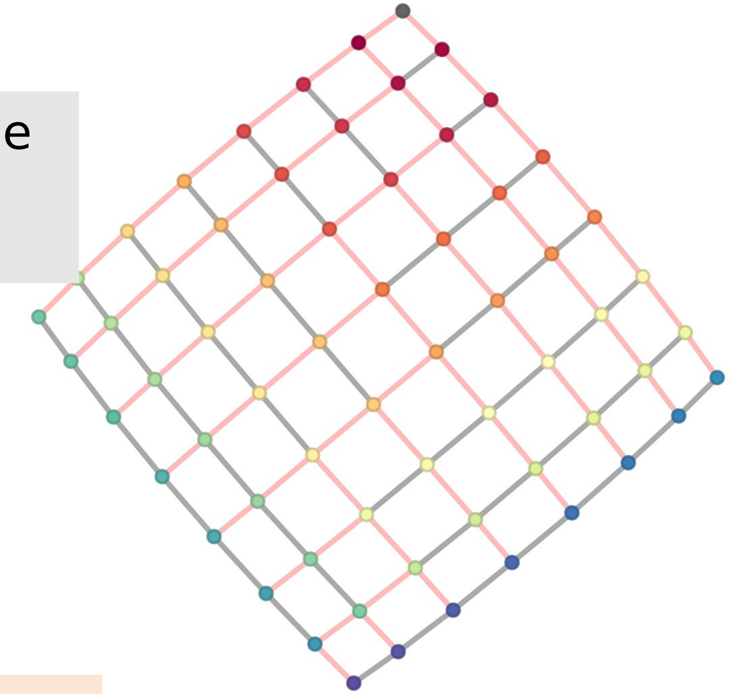
Loopy Local Gauges, *Weighted Model counting,* & big TNs

TNS subgroup 8th Oct 2020

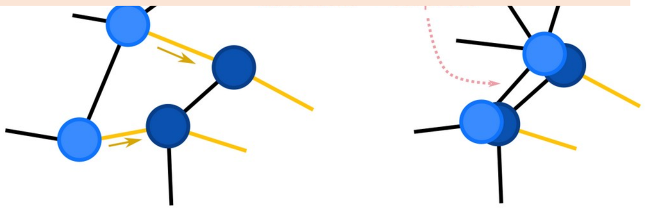
Arbitrary Approximate Contraction



Balanced Tree Compressed Contraction



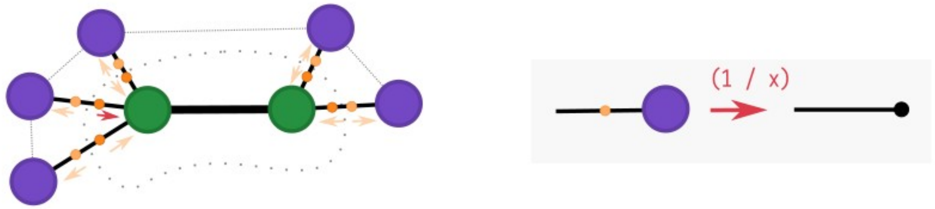
Spanning Tree Contract Around



Both require compression between ‘branches’ as we contract

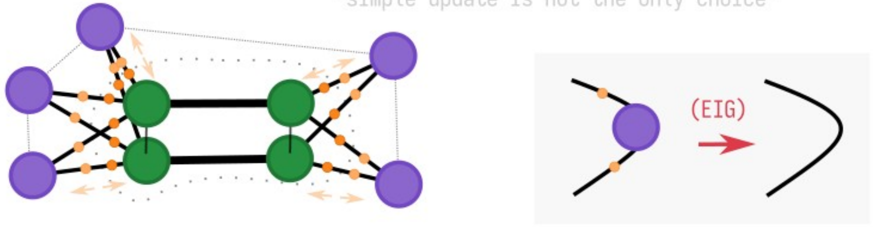
Gauging Strategies

scalar TN product gauge



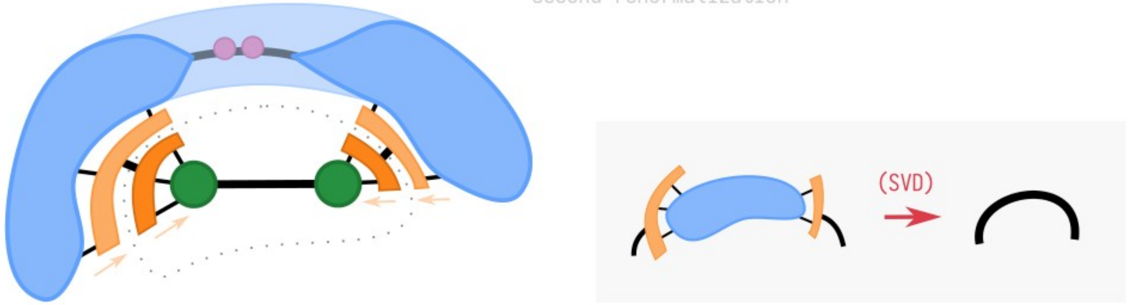
vector TN product gauge

"simple update is not the only choice"



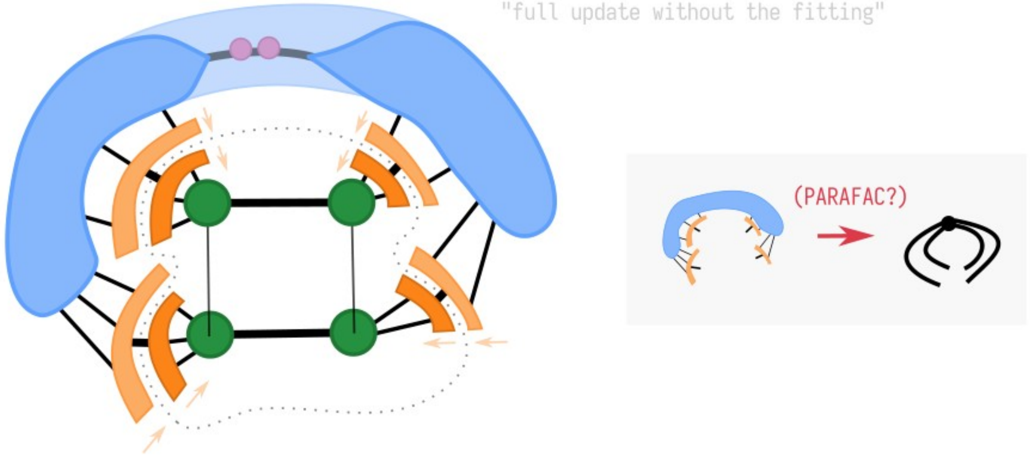
scalar TN full gauge (temporary)

"second renormalization"



vector TN full gauge (temporary)

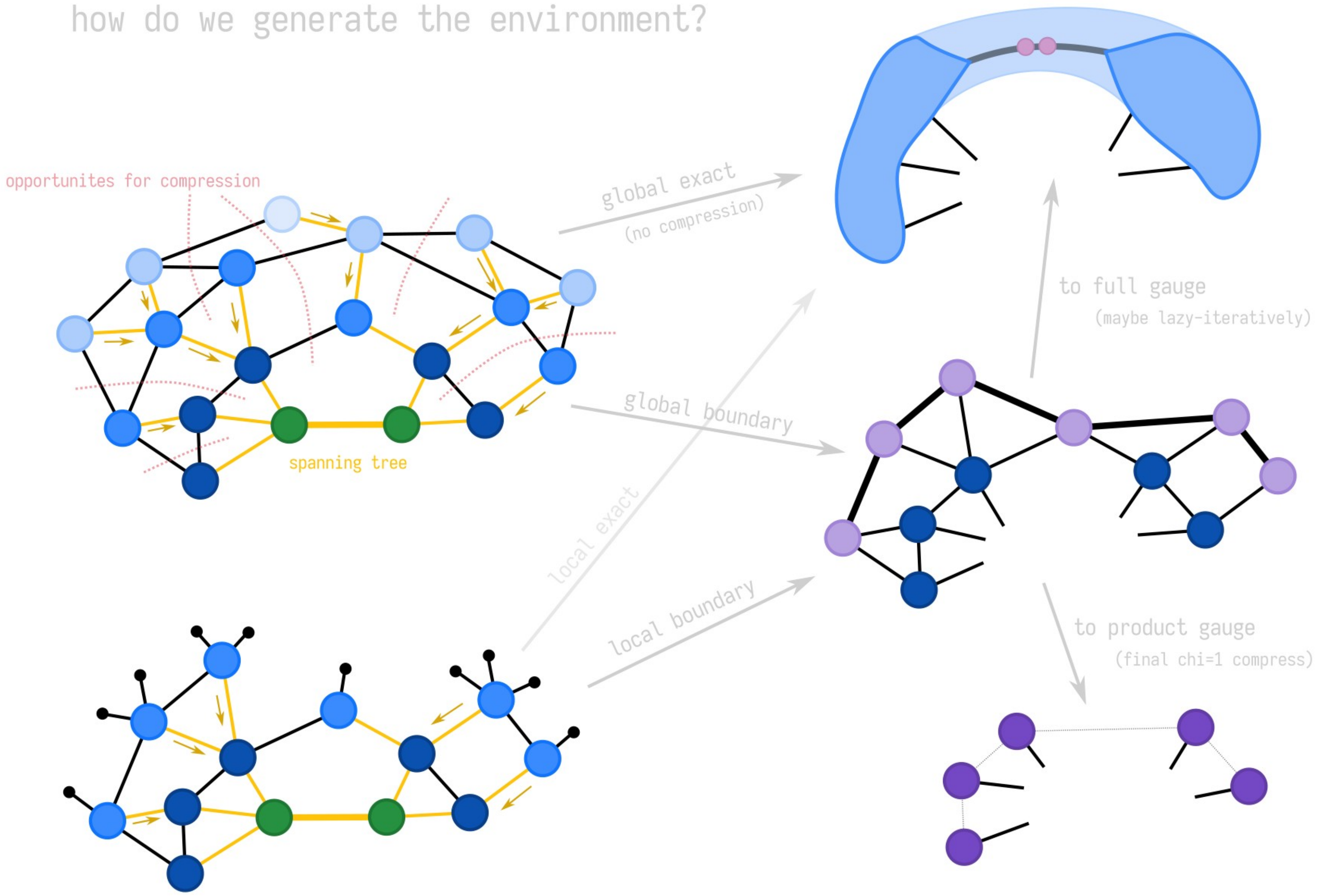
"full update without the fitting"



Local Environement

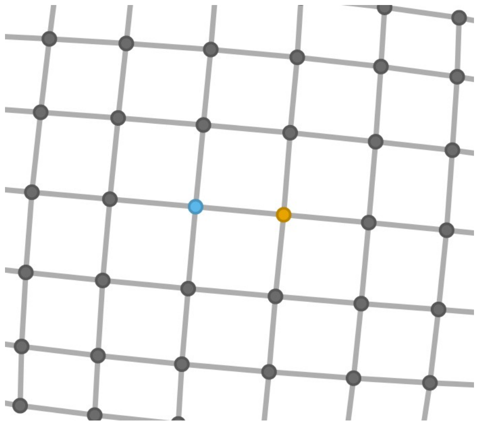
- Generating the local environment **is itself a compressed contraction**

- Albeit a smaller, easier one

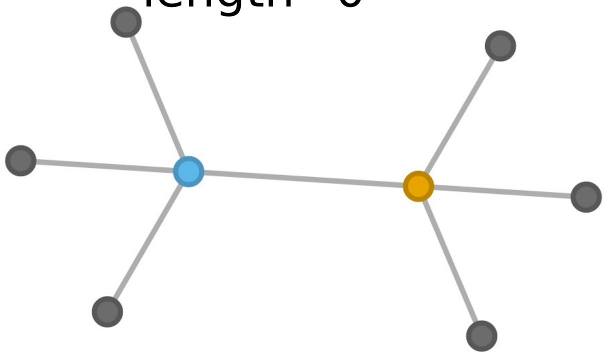


Adding Loops to LPG

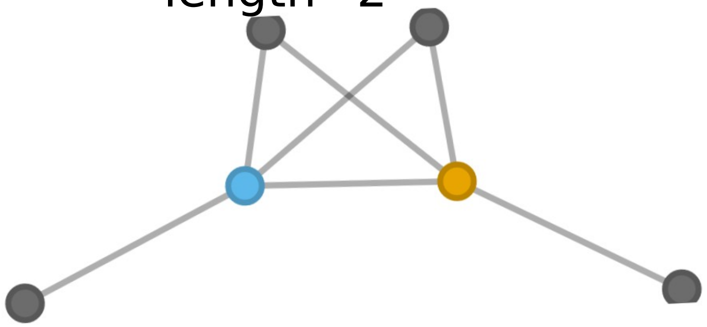
- In the vector LPG case, which we know works well, we try and diagonalize the effective environment into **identity loops**
  - **Should we be trying this in the scalar LPG case?**



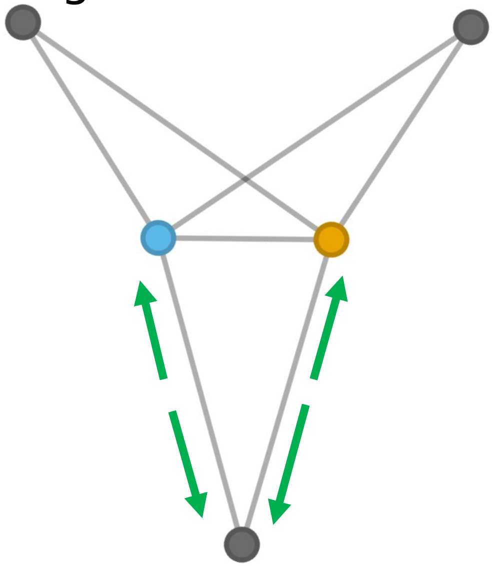
loop-length=0



loop-length=2



loop-length=8

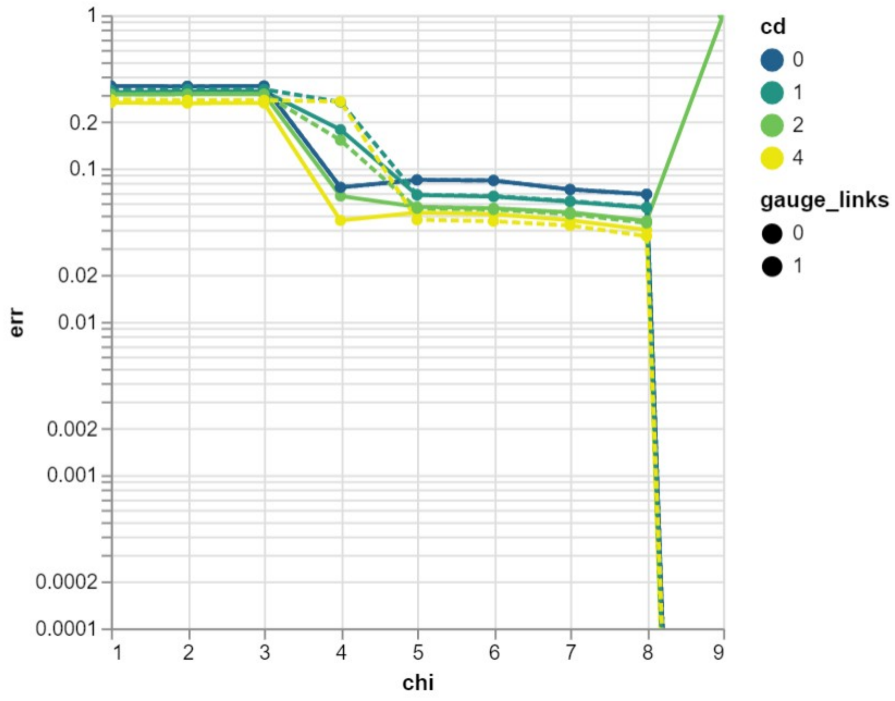


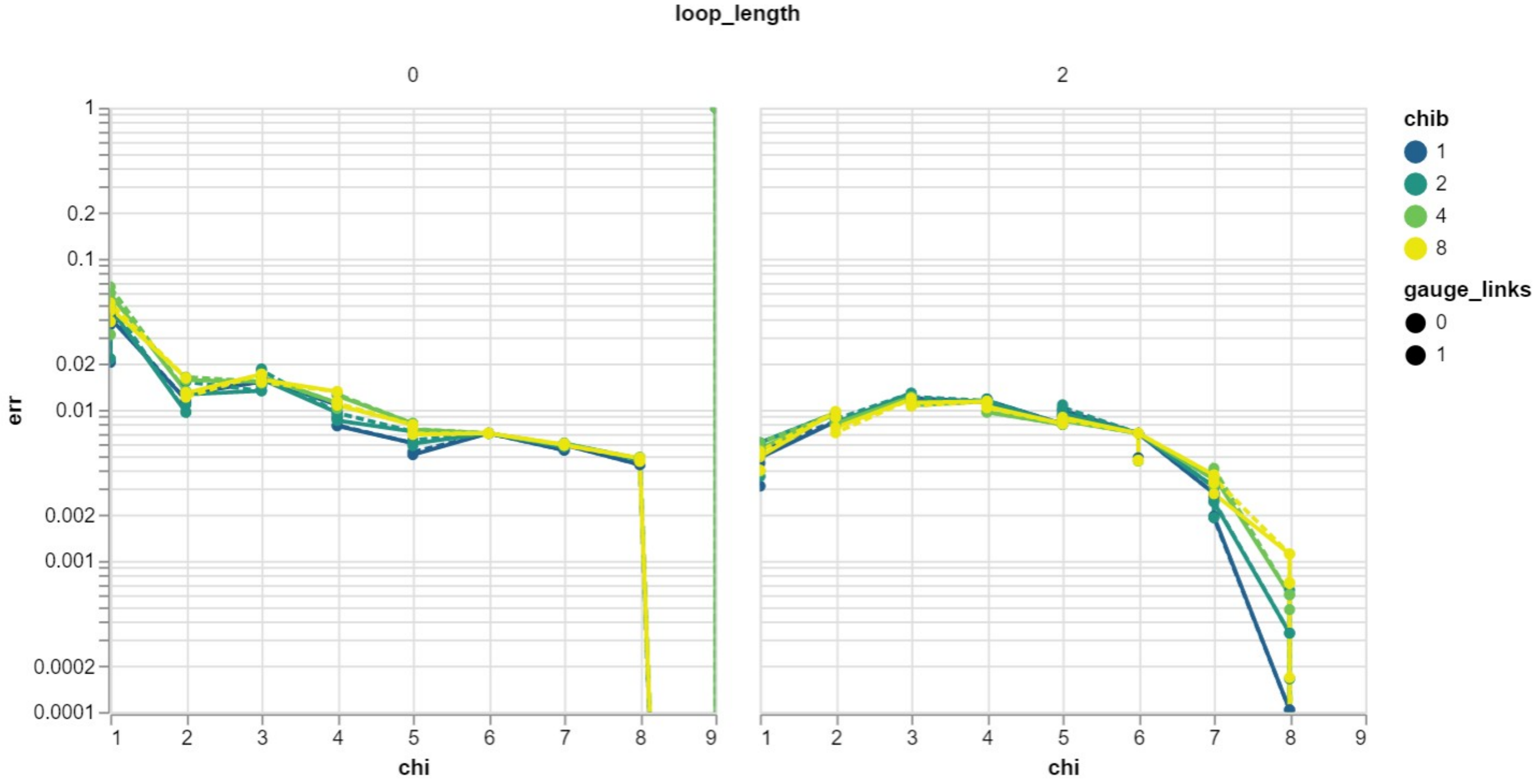
Gauge bonds with , etc

Gauge each env into identity

Works (very) well for Single Bonds

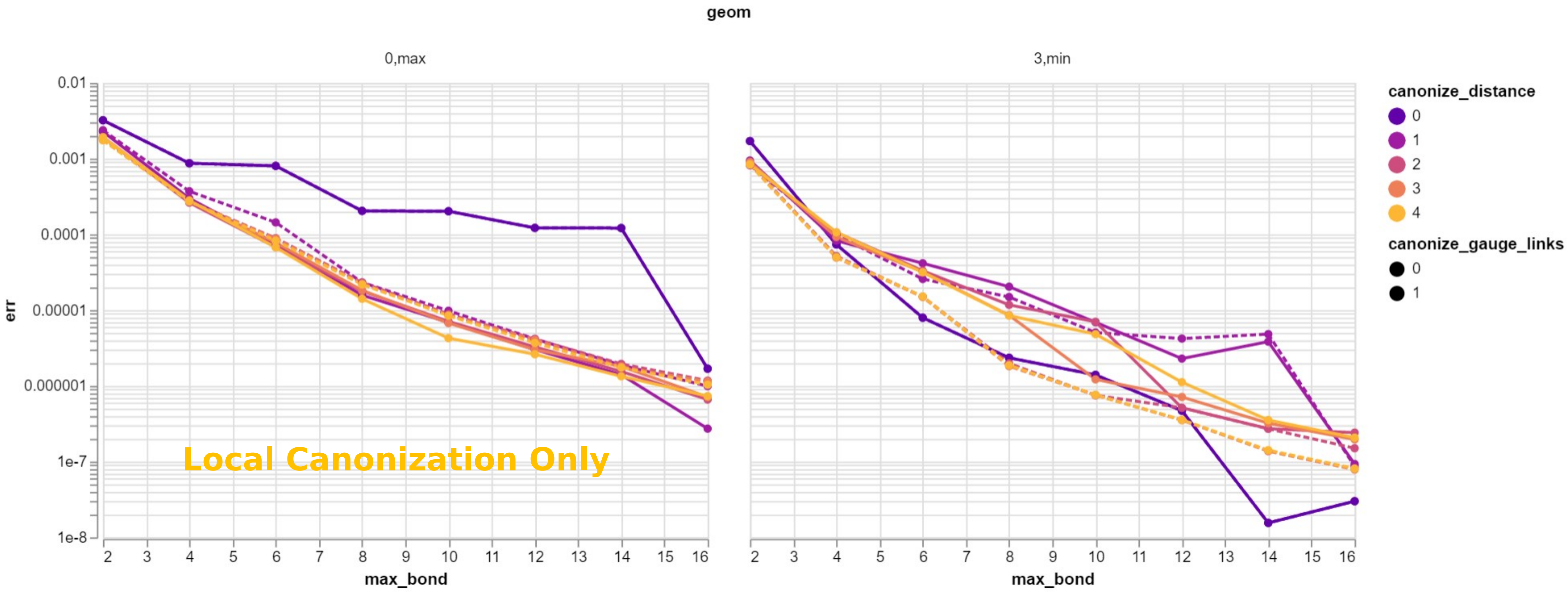
Error when compressing single bond of random 2D TN to `chi` after gauging:





But doesn’t seem to help general contractions

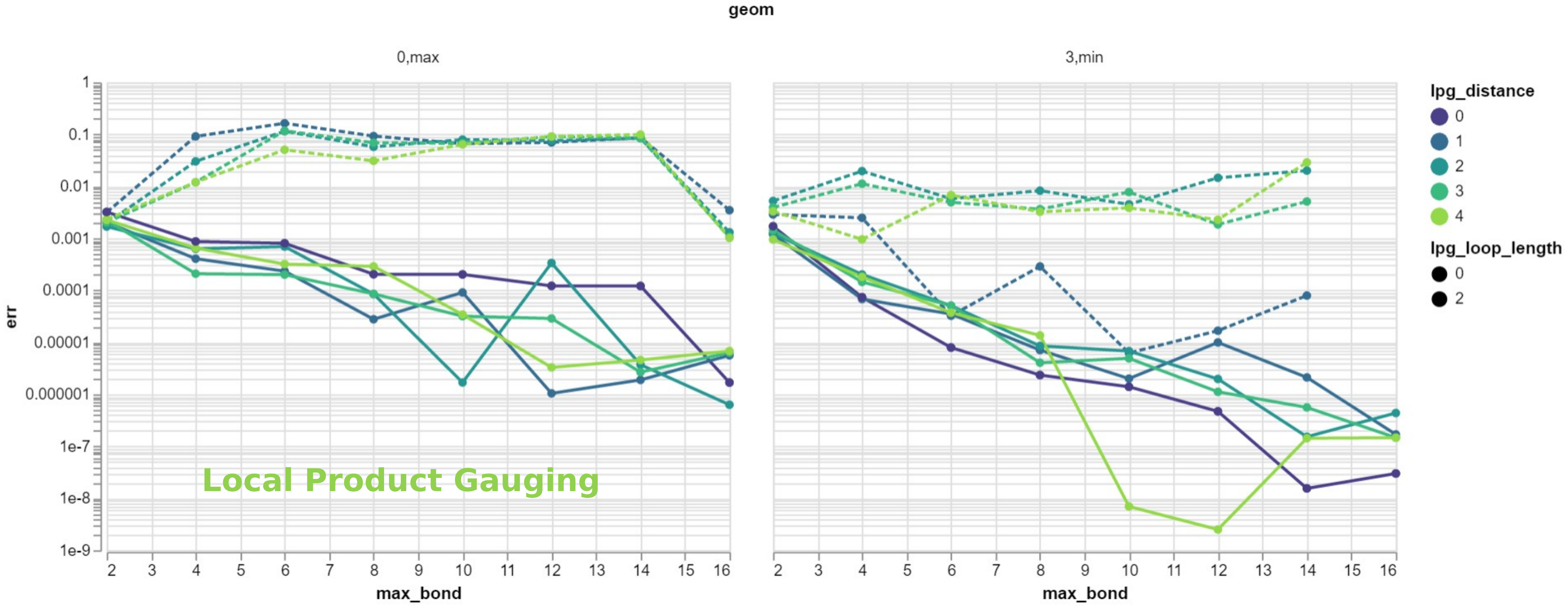
- Results on *5x5x5 ising partition function* ~critical point



Local Canonization Only

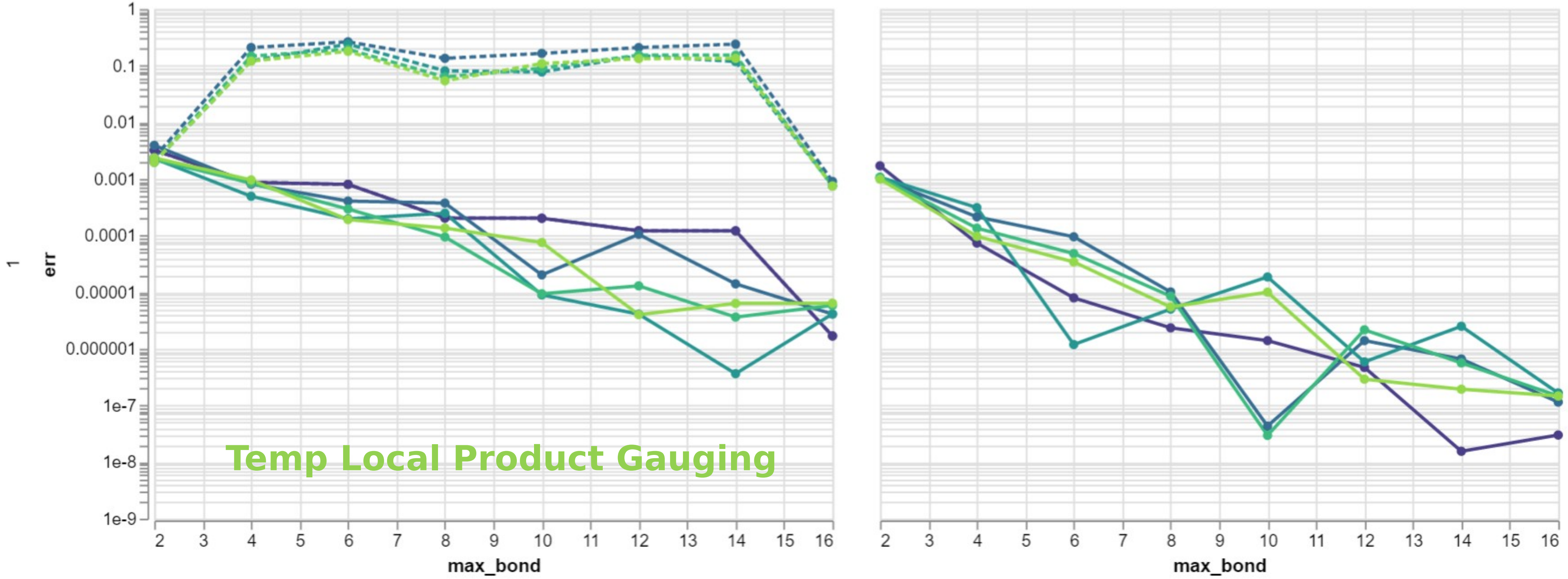
But doesn’t seem to help general contractions

- Results on *5x5x5 ising partition function* ~critical point



Could it be making other bonds harder? ‘Temporarily’ gauge only:

- Results on *5x5x5 ising partition function* ~critical point



Temp Local Product Gauging

(Weighted) Model Counting

SAT-isfying logical constraints:

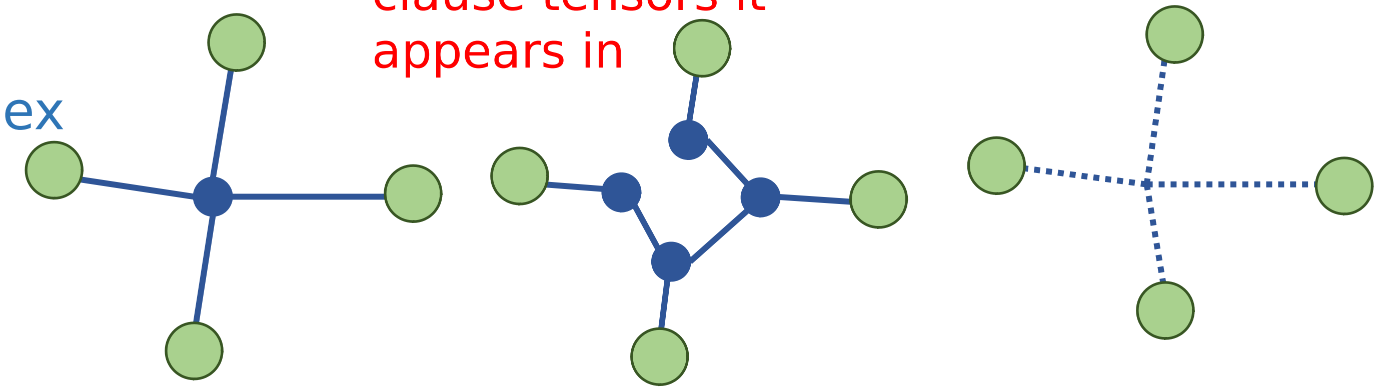
  - have some Boolean variables,
  - And list of clauses () ()

- **Model Counting** – number of solutions satisfying constraints
- ***Weighted* model counting** – each variable contributes weight

- **#P-Complete** -- i.e known that many hard problems map to this

Mapping to (hyper) tensor network

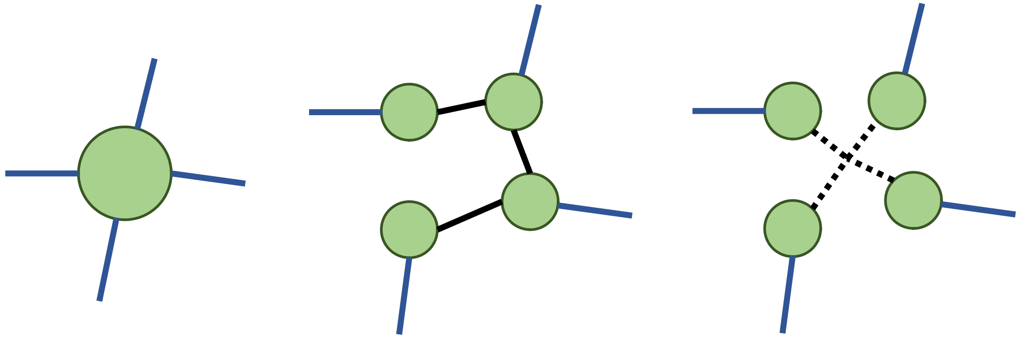
- Variable – tensor index
  - Usually one COPY-tensor per index
  - But we can use hyper-edges



Connected to clause tensors it appears in

Can use MPS or hyper-index to avoid high rank tensor

- Clause – OR tensor
  - All ones apart from single entry were no constraint is satisfied
  - Can either represent densely, or as MPS, or PARAFAC decomp



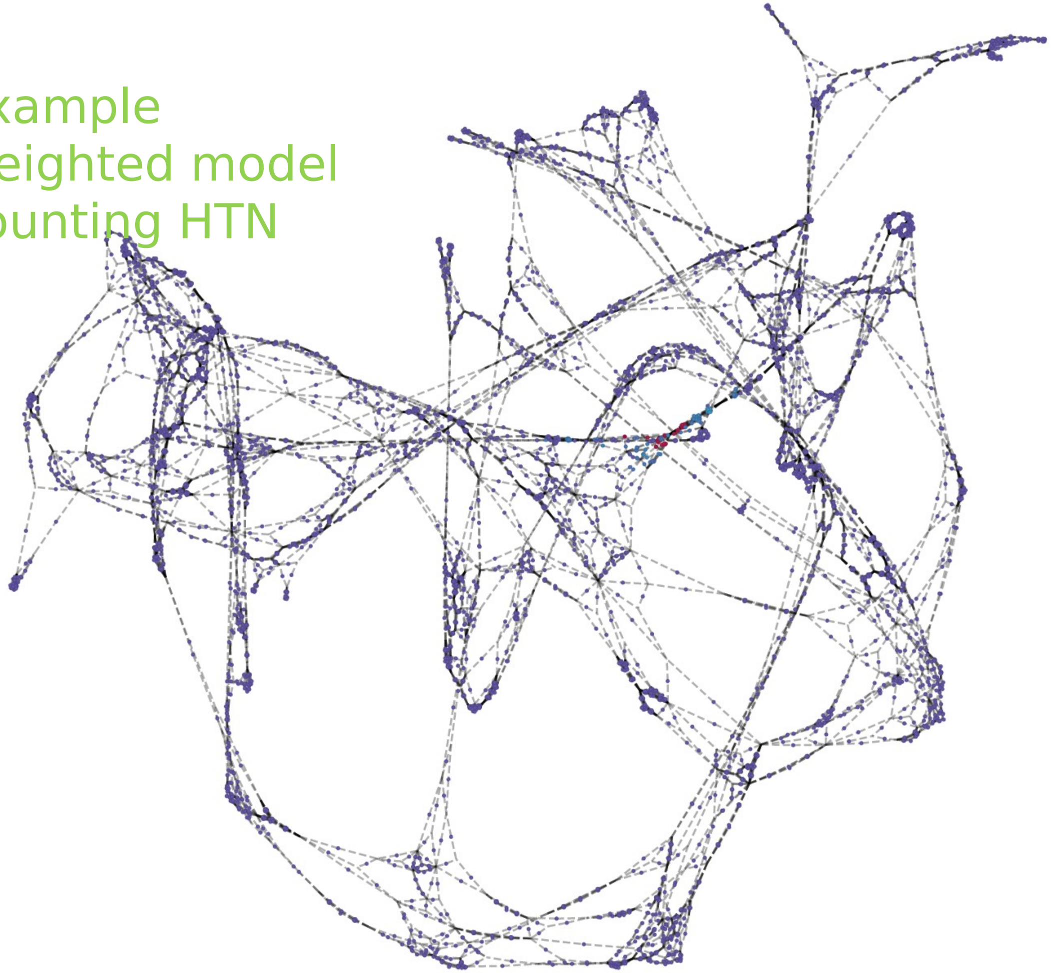
Works pretty well

On recent MC2020 competition:

Using
- hyper-edges
- local TN simplifications
- hyper optimized contraction

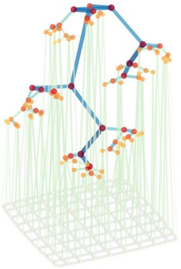
Solves 99 / 100 instances
(In actual competition winners only solved 69)

Example weighted model counting HTN

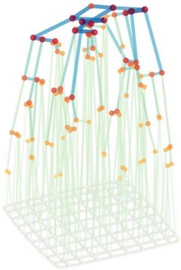


Hypergraph Centralities + Large TNs

- What is a centrality useful for?
  - Finding center of TN that we don’t know geometry of



  - Alternate way of defining surface and boundary contraction
  - Defining the contraction *order* of a balanced tree



  - Imbues a local direction to any graph that we can use:
    - e.g. when ‘dehypering’ a HTN, can use to order legs of an inserted MPS COPY-tensor
    - Can now do this on very large TNs + hyperTNs