BP - SU

The BP-SU algorithm consists of the following steps:

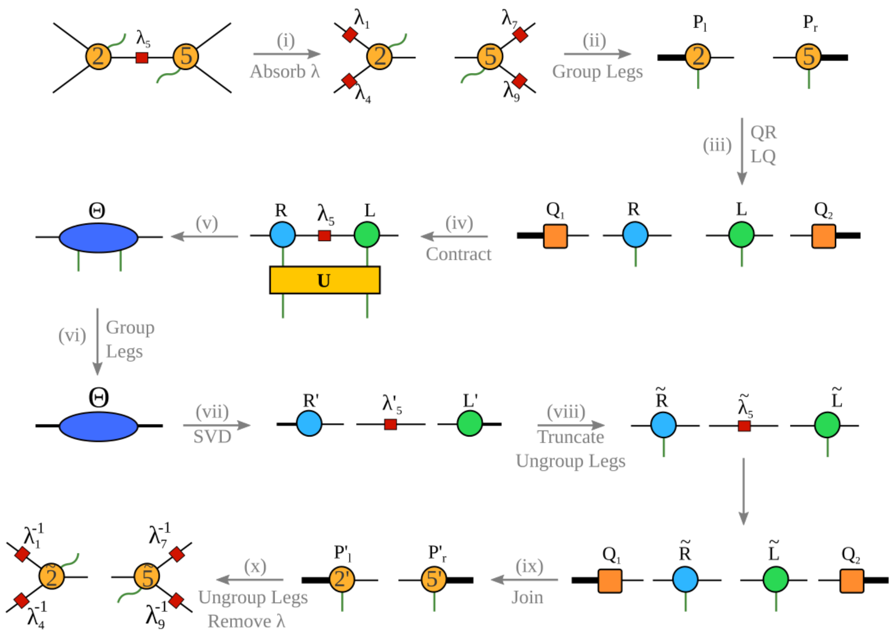
1. `qbp(T_list, e_list) => m_list`

2. `BP_gauging(T_list, e_dict, m_list) => T_list, w_dict`

3. ```
   apply_2local_gate(T_list, e_list, w_list, U_list)
       => T_list, w_dict
   ```

4. `merge_SU_weights(T_list, w_list) => T_list`
   ```

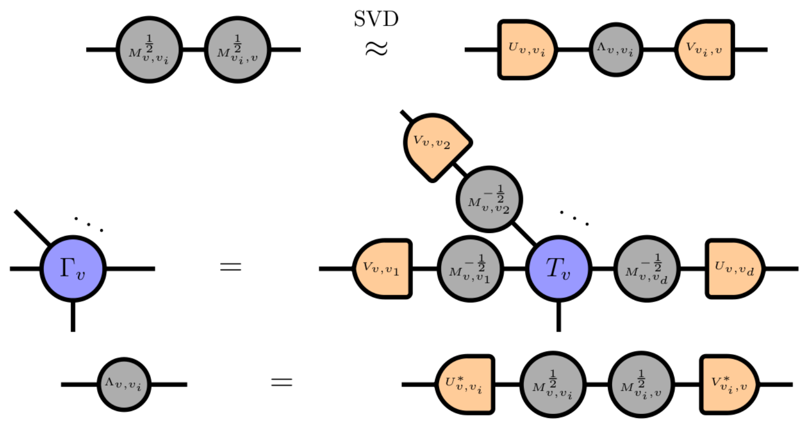
The Simple-Update procedure



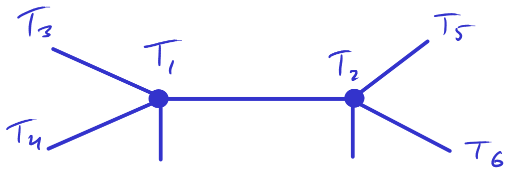
apply_2local_gate (T_i, T_j , ω_i, ω_j, ω, op, maxD, eps)

BP-Gauging

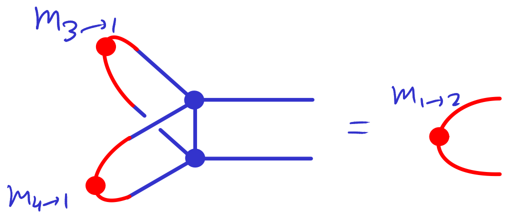
"Gauging tensor networks with belief propagation", Joseph Tindall and Matt Fishman, SciPost Phys. 15, 222 (2023)



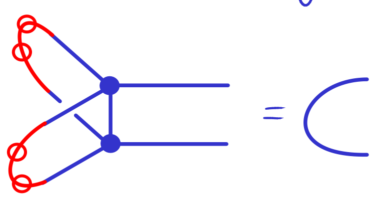
Explanation



The BP fixed points assure:

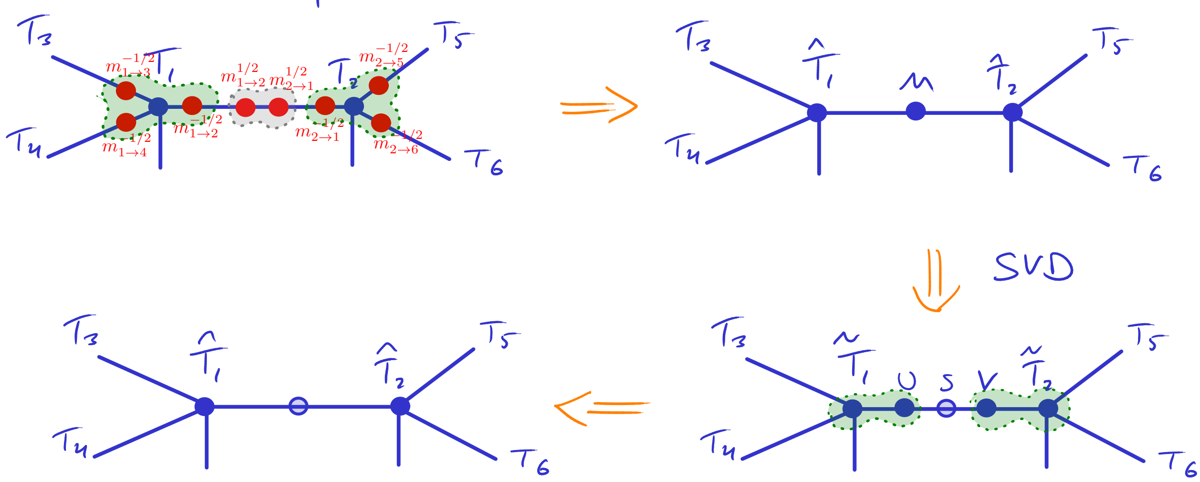


We want to take it to the Vidal Gauge:



We achieve this by:

d



The BP-Gauge fixing alg' should go like that:

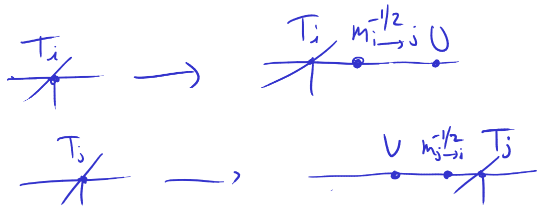
$\otimes$ Go over all edges $e = (i, j)$

1. Calc $m_{i\to j}^{\pm 1/2}$, $m_{j\to i}^{\pm 1/2}$

2. Calc $M_{ij} = m_{i\to j}^{1/2} \cdot m_{j\to i}^{1/2}$

3. SVD: $M_{ij} = U \cdot S \cdot V$

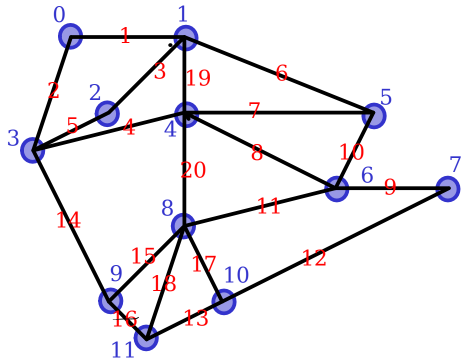
4. Absorb



Example on a random PEPS

The example-BP-SU.py file contains an example of using the BPSU library for running SU on a random PEPS (on some random graph), and using BP to move to the Vidal gauge.

The structure of the PEPS is as follows:



Note that blue labels denote the vertex number and red labels denote the edge labels