Active physical support compression in the CPSV16 commuting-form argument

2026-07-30

Verdict: scope-restriction (open).

1 Source passage

In the standing Case I context, $\mathcal{K}$ is an injective normal tensor generating matrix product density operators. Lemma `propSN` of Cirac–Pérez-García–Schuch–Verstraete[1] assumes in addition that $\mathcal{K}$ fulfills the saturation of the area law (SAL), and concludes that there exist an isometry U and tensors l_k, r_k such that

$$U\mathcal{K}_{\beta,\alpha}U^* = \bigoplus_k (l_k)_\beta \otimes (r_k)_\alpha$$

with

$$\eta_{k,h} = \sum_a (r_k)_a \otimes (l_h)_a \geq 0,$$

and the non-negative matrix $T_{k,h} = \operatorname{tr}(\eta_{k,h})$ is primitive.

The passage defines U, r_k, l_k, and $\eta_{k,h}$ at lines 1436–1450. It defines $T_{k,h}$ at lines 1452–1455 and proves its primitivity through line 1471. It does **not** construct the active physical support compression: choosing isometries onto the joint column supports of the left and right factor families per sector, giving inactive sectors zero-dimensional coordinate spaces, and assembling the compressed injective tensor.

Proposition `3to4`, at lines 1570–1594, uses the uncompressed neighboring operators on the full factor spaces to form commuting positive bonds. The later Proposition `prop3to4`, at lines 1786–1797, concerns a simple tensor with a basis of normal tensors and invokes the preceding injective result. These are distinct source statements.

[1]Source: `Papers/1606.00608/MPDO-22-12-17-2.tex:1374--1471`.

2 The compression construction

The four files `TNLean/MPS/MPDO/PhysicalSectorActiveFactorSupport`, `TNLean/MPS/MPDO/PhysicalSectorActiveRestriction`, `TNLean/MPS/MPDO/PhysicalSectorActiveNeighboring`, and `TNLean/MPS/MPDO/PhysicalSectorCoordinateTransport` build the following auxiliary construction after Lemma `propSN` (lines 1406–1471). It is used in the formal argument corresponding to Proposition `prop3to4` (line 1787):

1. **Factor-support coordinates:** For each active sector, choose isometries V_k^L, V_k^R whose ranges are the joint column supports of the left and right factor families $\{l_{k,\beta}\}_\beta$ and $\{r_{k,\alpha}\}_\alpha$. For inactive sectors, set both coordinate dimensions to zero. The product range of each pair is exactly the family-support projection of the sector product family. Injectivity of the original tensor and positivity of every neighboring operator imply the adjoint-closure relation used to prove two-sided absorption by these support projections.

2. **Compressed factors:** Form $\tilde{l}_{k,\beta} = (V_k^L)^\dagger l_{k,\beta} V_k^L$ and similarly for $\tilde{r}_{k,\alpha}$. These are the factor matrices restricted to the active coordinate spaces.

3. **Physical inclusion:** Assemble a block-diagonal isometry V from the sector support isometries, transport it back to original physical coordinates via $U^\dagger$, and obtain $V : \mathbb{C}^e \to \mathbb{C}^d$ with $VV^\dagger$ equal to the canonical joint column support of the original physical slices.

4. **Compressed tensor:** Define $K_{\text{act}} = V^\dagger K V$. This tensor lives on $\mathbb{C}^e$ where e is the total dimension of the retained active-factor supports. Its injectivity follows from the injectivity of K together with positivity of every neighboring operator.

5. **Neighboring congruence:** The compressed neighboring operator is the congruence of the ambient one by the Kronecker product of the source right and target left support isometries. Positive semidefiniteness follows from preservation under matrix congruence.

6. **Coordinate transport:** The generic operation $M \mapsto VMV^\dagger$ changes physical coordinates of any MPO tensor, together with its two-site and four-site closures.

None of these steps appears in lines 1406–1471 of CPSV16. The source has the block factorization and the η construction; the compression removes inactive dimensions as an auxiliary input preparation in the formal route to the theorem corresponding to Proposition `prop3to4`. Under injectivity of the original tensor and positivity of the neighboring operators, the joint column support projections absorb every physical slice on both sides. Compression by their conjugate transposes therefore preserves exact two-sided recovery.

3 Affected declarations

The nine load-bearing declarations are the choice of support isometries for an active sector,[2] the simultaneous choice of factor-support coordinates,[3] the compressed left factor,[4] recovery of the original physical slice from sector coordinates,[5] the restricted tensor,[6] compression of the sectorwise factorization by the joint support inclusion,[7] exact recovery after returning to the original physical coordinates,[8] injectivity of the restricted tensor,[9] and the assembled restriction datum.[10]

4 Blueprint dependency

The chain `PhysicalSectorActiveRestriction` → `PhysicalSectorActiveNeighboring` → `PhysicalSectorActiveBond` → `BNTPhysicalSectorGSNNCH` feeds the `\leanok`-tagged capstone `MPOTensor.hasGSNNCHForm_of_bntLayerOrthogonal_of_physicalSectorFactorization` labelled `thm:mpdo_bnt_physical_sector_gsnnch` in `blueprint/src/chapter/ch21_mpdo_rfp_simple_local_structure_capstone.tex`. The compression is mathematically sound but the source-citation chain is incomplete without the markers added in this note.

Classification: scope restriction. The compression itself is mathematically verified. The local development first passes to the canonical active physical support and states the coordinate changes separately, while the source propositions do not contain these constructions. The markers added by this disclosure make that distinction explicit.

[2] `MPOTensor.PhysicalSectorFactorization.exists_activeSector_factorSupportIsometries`.

[3] `MPOTensor.PhysicalSectorFactorization.ActiveFactorSupportData`.

[4] `MPOTensor.PhysicalSectorFactorization.compressedLeftTensor`.

[5] `MPOTensor.PhysicalSectorFactorization.physicalSlice_eq_conjTranspose_mul_sectorCoordinateTensor_mul`.

[6] `MPOTensor.PhysicalSectorFactorization.activePhysicalSupportRestriction`.

[7] `MPOTensor.PhysicalSectorFactorization.dependentSupportInclusion_compression`.

[8] `MPOTensor.PhysicalSectorFactorization.changePhysicalBasis_physicalSupportInclusion_activePhysicalSupportRestriction`.

[9] `MPOTensor.PhysicalSectorFactorization.activePhysicalSupportRestriction_isInjective`.

[10] `MPOTensor.PhysicalSectorFactorization.activePhysicalSupportRestrictionData`.