Injective PEPS Fundamental Theorem: Section 3 Route

2026-05-06

Verdict: clarification (resolved).

1 The source argument

The injective PEPS part of [MGRPG+18] proves the Fundamental Theorem for tensor networks on finite graphs without double edges and without self-loops. The proof is local. It does not slice a two-dimensional network into a long one-dimensional chain. Instead, it fixes an edge, blocks the graph around that edge into a three-site injective MPS, applies the one-dimensional injective theorem, and repeats this construction over all edges.[1]

For a chosen edge $e = (u, v)$, all vertices other than u and v are grouped into the middle tensor. The two endpoint tensors and the middle tensor form a three-site MPS. Injectivity of the PEPS tensors implies injectivity of this blocked MPS. Since the two original PEPS represent the same state, the two blocked MPS represent the same state. The one-dimensional injective theorem therefore gives an algebra isomorphism between matrix insertions on the chosen bond. Equivalently, an arbitrary matrix X inserted on that bond in one network is matched by a matrix on the corresponding bond in the other network.

After choosing the resulting edge gauges for every edge and absorbing them into one tensor family, the paper obtains the following stronger comparison: for every edge and every matrix X, inserting X on that edge in the first PEPS gives the same state as inserting X on the same edge in the modified second PEPS.[2]

2 Current formal representation

The finite graph, edge, incident-edge, tensor, virtual configuration, state coefficient, equality of states, and vertex injectivity are represented in `TNLean/PEP`

[1] The edge-blocking construction is `Papers/1804.04964/paper_normal.tex:981--1009`. The equality of the two PEPS before blocking is at `Papers/1804.04964/paper_normal.tex:1010--1036`. The application of the three-site injective MPS theorem is at `Papers/1804.04964/paper_normal.tex:1037--1065`.

[2] This is the displayed equation labelled `eq:inj_equal_edge` in `Papers/1804.04964/paper_normal.tex:1037--1065`.

S/Defs.lean. Vertex injectivity is the linear independence of the local physical vectors indexed by local virtual configurations, matching the paper's statement that each tensor is injective as a map from the virtual space to the physical space.

The local linear-algebraic realization of virtual operations at an injective vertex is represented in TNLean/PEPS/VirtualInsertion.lean. The chosen left inverse of the local tensor map and the resulting physical realization of local virtual endomorphisms correspond to the same observation used in the three-site MPS proof: a virtual operation can be represented by a physical operation on a neighboring site. The basis computation TNLean.PEPS.localIncidentMatrixOp_single and its tensor-map form TNLean.PEPS.localTensorMap_localIncidentMatrixOp_single identify the one-edge matrix action on a single local virtual boundary with the expected sum over the new distinguished edge index.

The edge-centered blocking data are represented in TNLean/PEPS/Blocking.lean. The existing declarations split local virtual configurations at a distinguished incident edge and split the vertex product into the left endpoint, the middle region, and the right endpoint. The edge-blocked coefficient is proved equal to the original PEPS coefficient.

The next formal layer is the matrix-insertion comparison. The declarations TNLean.PEPS.EdgeInsertedBoundaryConfig, TNLean.PEPS.edgeBoundaryToInsertedBoundaryConfig, TNLean.PEPS.EdgeDiagonalInsertedBoundaryConfig, TNLean.PEPS.edgeBoundaryConfigEquivDiagonalInsertedBoundaryConfig, TNLean.PEPS.edgeMiddleConfigToOpenMiddleConfig, TNLean.PEPS.edgeOpenMiddleConfigToMiddleConfig, TNLean.PEPS.edgeMiddleConfigEquivOpenMiddleConfig, TNLean.PEPS.edgeOpenMiddleWeight, TNLean.PEPS.edgeBoundaryLeftIndexEquivInsertedBoundaryConfig, TNLean.PEPS.edgeBoundaryRightIndexEquivInsertedBoundaryConfig, and TNLean.PEPS.edgeInsertedCoeff represent the open edge contraction used when an arbitrary matrix is inserted on the distinguished edge. These definitions are the formal counterpart of the matrix insertion appearing before and after the equation labelled eq:inj_equal_edge. The two maps TNLean.PEPS.edgeMiddleConfigToOpenMiddleConfig and TNLean.PEPS.edgeOpenMiddleConfigToMiddleConfig have been assembled into the finite equivalence TNLean.PEPS.edgeMiddleConfigEquivOpenMiddleConfig. This isolates the reindexing needed for the identity-insertion specialization. The current Lean layer includes the corresponding middle-tensor reindexing theorem TNLean.PEPS.edgeMiddleWeight_eq_edgeOpenMiddleWeight. The diagonal summand has also been isolated in TNLean.PEPS.edgeInsertedCoeff_identity_diagonal_summand, and TNLean.PEPS.edgeInsertedCoeff_identity_offDiagonal_summand records the pointwise off-diagonal vanishing. The full identity-insertion specialization is TNLean.PEPS.edgeInsertedCoeff_identity; its proof uses TNLean.PEPS.edgeBoundaryConfigEquivDiagonalInsertedBoundaryConfig to reindex the finite inserted-boundary sum along the diagonal.

3 Formalized Section 3 route

The following statements record how the formal proof follows the Section 3 route.[3] They form a status account: each item names the formal statement that carries the corresponding mathematical step.

1. Vertex injectivity passes to the edge-blocked three-site object. This is the formal version of the paper's assertion that the regrouped MPS is injective. The middle physical index for this regrouped object is now explicit as `TNLean.PEPS.EdgeMiddlePhysicalConfig`, and the middle contractions have the middle-indexed forms `TNLean.PEPS.edgeMiddleWeightOn` and `TNLean.PEPS.edgeOpenMiddleWeightOn`. In the blueprint the injectivity statement is `TNLean.PEPS.EdgeBlockedThreeSiteInjective`. The singleton base case is now formalized by `TNLean.PEPS.singletonRegionTensorFamily` and `TNLean.PEPS.IsVertexInjective.singletonRegionTensorInjective`: for a one-vertex block $\{v\}$, $T_{A,\{v\}}(\eta,\sigma) = A_v(\eta,\sigma)$, hence vertex injectivity gives $\mathrm{inj}(T_{A,\{v\}})$. The passage to the abstract region-injectivity predicate is recorded by `TNLean.PEPS.SingletonRegionInjectivityComparison` and `TNLean.PEPS.SingletonRegionInjectivityFromVertexInjectivity.ofSingletonComparison`. With the abstract union-closure lemma, this gives every nonempty finite region by `TNLean.PEPS.RegionInjectivityUnionClosure.finset_injective_of_singletons`: $R = \bigcup_{v\in R}\{v\}$ and $\kappa(\{v\})$ for every $v \in R$ imply $\kappa(R)$. The corresponding nonempty-middle edge-middle tensor consequence is recorded in `thm:peps_edgeMiddleTensorInjective_of_singletons`: for $M_e = V \setminus \{u,v\}$, the assumptions $M_e \neq \varnothing$ and $M_e = \bigcup_{w\in M_e}\{w\}$ give $\kappa(M_e)$; the edge-middle comparison then gives injectivity of the middle tensor. The endpoint reduction then gives the one-edge three-site statement `thm:peps_edgeBlockedThreeSiteInjective_of_singletons`, and applying this to all edges satisfying $M_e \neq \varnothing$ gives `thm:peps_edgeBlockedThreeSiteInjective_all_of_singletons`. The graph-cardinality variant `TNLean.PEPS.EdgeMiddleRegionInjectivityComparison.edgeBlockedThreeSiteInjective_all_two_lt_card` covers the common case $2 < |V|$: since $|M_e| = |V| - 2 > 0$ for every edge, the nonempty-middle hypothesis is automatic. Vertex injectivity supplies the two endpoint assertions by `TNLean.PEPS.IsVertexInjective.edgeBlockedEndpointTensorMaps_injective` and, for all edges at once, by `TNLean.PEPS.IsVertexInjective.edgeBlockedEndpointTensorMaps_injective_all`, and `TNLean.PEPS.IsVertexInjective.edgeBlockedThreeSiteInjective_of_middle` reduces the proof to the middle-block injectivity statement. Its all-edge form is `TNLean.PEPS.IsVertexInjective.edgeBlockedThreeSiteInjective_all_of_middle`. The comparison `TNLean.PEPS.EdgeMiddleRegionInjectivityComparison` records the conditional passage from finite-region injectivity of $V \setminus \{u,v\}$

[3] This list and the ledger below are the status record referenced by the Section 3 docstrings and blueprint comments.

to the edge-middle tensor family. Its theorem `TNLean.PEPS.EdgeMiddleRegionInjectivityComparison.edgeBlockedThreeSiteInjective` then combines this comparison with vertex injectivity to obtain the edge-blocked three-site assertion at one edge. The all-edge middle-tensor comparison is `TNLean.PEPS.EdgeMiddleRegionInjectivityComparison.edgeMiddleTensorInjective_all`. The all-edge conditional form is `TNLean.PEPS.EdgeMiddleRegionInjectivityComparison.edgeBlockedThreeSiteInjective_all`. The finite induction on exposed middle vertices is separated as `TNLean.PEPS.FiniteRegionKernelDescent`: for kernel conditions $K_c(S)$ attached to a coefficient family $c = (c_\rho)_\rho$, the implications $K_c(S) \Rightarrow K_c(S \setminus \{j\})$ imply $K_c(M_e) \Rightarrow K_c(\varnothing)$. The edge-middle instance of this coefficient argument is recorded by `TNLean.PEPS.EdgeMiddleKernelDescentData`: the zero relation $\sum_\rho c_\rho T^\rho_{A,M_e}(\tau) = 0$ for every middle physical index τ gives $K_c(M_e)$, and $K_c(\varnothing)$ gives $c_\rho = 0$ for every ρ. Consequently `TNLean.PEPS.EdgeMiddleKernelDescentData.edgeMiddleTensorInjective` proves $\mathrm{inj}(T_{A,M_e})$ from such a datum, and `TNLean.PEPS.IsVertexInjective.edgeBlockedThreeSiteInjective_of_kernelDescent` combines it with the endpoint injectivity assertions. The implication from vertex injectivity to middle-block injectivity is false without a positivity hypothesis on the virtual bond dimensions. If some complement bond makes `TNLean.PEPS.EdgeComplementConfig` empty, then the middle tensor family is identically zero (`TNLean.PEPS.edgeMiddleTensorFamily_eq_zero_of_isEmpty`), and the family is not linearly independent when the boundary-label index is nonempty (`TNLean.PEPS.not_edgeMiddleTensorInjective_of_isEmpty`). A four-cycle with a zero-dimensional middle-to-middle edge gives the corresponding mathematical example in which vertex injectivity can survive vacuously at vertices incident to that bond. The source avoids this because injective PEPS have nonzero-dimensional virtual bond spaces. The hypothesis $\forall f,\ 0 < \dim(f)$ is therefore part of `TNLean.PEPS.IsVertexInjective.edgeBlockedThreeSiteInjective`; the two lemmas above record the formally checked non-injectivity obstruction.

The mathematical step represented by the restored theorem is the contraction theorem used in the paper: contracting vertex-injective tensors over that finite middle region gives an injective blocked tensor. The source proves this in one step from the one-sided inverse, where the inverse of a contraction of injective tensors is the contraction of the inverses up to the connecting bond dimension (`Papers/1804.04964/paper_normal.tex`, lines 205–250). The kernel descent recorded above is the finite-induction route towards that one-step fact, not a separate argument in the source. The full transfer is stated as `TNLean.PEPS.IsVertexInjective.edgeBlockedThreeSiteInjective` with the positive-bond hypothesis; its proof consists of the three facts below.

The per-vertex one-sided-inverse step is expressed by `TNLean.PEPS.IsVertexInjective.localCoeff_eq_zero_of_contract_zero`: a coefficient

family on the local virtual configurations at a vertex v that contracts to the zero physical vector vanishes. The middle tensor is a genuine per-vertex product, $T_{A,M_e}(\rho,\tau) = \sum_\zeta \prod_{v\in M_e} A_v(\zeta,\tau_v)$, so a single vertex j is removed by $\prod_{v\in M_e} = A_j \cdot \prod_{v\in M_e\setminus\{j\}}$. Three facts instantiate the kernel descent from vertex injectivity:

(a) the star-bond split equivalence `TNLean.PEPS.edgeComplementConfigSplitAt`, decomposing the complement bond configuration into the star bonds of j (its local virtual configuration) and the remaining bonds, with a first-component law matching edgeComplementValue; the shared bond between two middle vertices is read back from the star factor, which is why the peeled coefficient genuinely depends on j's local virtual configuration;

(b) the partial-contraction kernel condition over a finite open region S together with its one-vertex deletion $K_c(S) \Rightarrow K_c(S\setminus\{j\})$, obtained by fixing the exposed bonds, ranging the open physical leg of j, and applying `TNLean.PEPS.IsVertexInjective.localCoeff_eq_zero_of_contract_zero`;

(c) the terminal isolation $K_c(\varnothing) \Rightarrow c = 0$ over the residual endpoint boundary labels, by surjectivity of the boundary labelling, which is where the positive bond dimensions are used.

These assemble into `TNLean.PEPS.edgeMiddleKernelDescentData` and close the target by the existing reductions.

2. The two internal steps in Lemma `inj_isomorph` are formalized for the edge-blocked three-site MPS. The local endpoint part of the virtual-to-physical realization $X \mapsto O_1, O_2$ is formalized by `thm:peps_virtualInsertionPhysicalRealization`: vertex injectivity turns the one-leg virtual matrix action at either endpoint into a physical operator on that endpoint. The endpoint coefficient-level direction is formalized by `thm:peps_edgeInsertedCoeff_endpointPhysicalRealization`: after fixing ordinary edge-boundary data, the left and right physical operators supply the independent inserted-edge index at the corresponding endpoint. On the left endpoint the formal statement uses the transpose of the inserted matrix, because the ordinary boundary supplies the right bond index. The source recovery $O_1, O_2 \mapsto W$ is proved as `TNLean.PEPS.physical_to_virtual_insertion`. The middle block is contracted out by `TNLean.PEPS.resonate_middle_inverted`, leaving the bond-contracted endpoint identity. Inverting the right endpoint (`TNLean.PEPS.resonate_invert_right_endpoint`) reads the common matrix M off the action of O_1; inverting the left endpoint (`TNLean.PEPS.resonate_invert_left_endpoint`) reads the action of O_2 off as a bond matrix, and `TNLean.PEPS.resonate_endpoint_coeff_reconcile` forces the two to agree, which is the source step $V = W$. The positivity hypothesis on the virtual bond dimensions supplies the reference residual configurations

used to fix M. Its local endpoint recovery component is also formalized by `TNLean.PEPS.edgeEndpointLocalVirtualOpOfPhysicalOp_eq_of_projected_realization_eq`: once the compressed endpoint actions are known to realize the same bond matrix, the local virtual pullbacks recover that matrix on the shared bond. The image-preserving conditional form `TNLean.PEPS.edgePhysicalToVirtualInsertion_of_projected_realization_eq` records the endpoint conclusion under explicit projected-realization and image-preservation hypotheses.

3. Equality of PEPS states is upgraded to equality of edge-inserted coefficients after the three-site reduction. This is the step that produces the matrix-algebra isomorphism on the chosen bond. The inserted-coefficient correspondence $X \mapsto Y$ is now formalized by `TNLean.PEPS.exists_edgeInsertedCoeff_eq`: for every inserted matrix X on the chosen bond of the first blocked PEPS there is a matrix Y on the corresponding bond of the second blocked PEPS with equal edge-inserted coefficients at every physical configuration. Its proof rests on the state and endpoint-operator comparison `TNLean.PEPS.edgeRealizationSum_left_eq_sum_edgeBlockedCoeff` and `TNLean.PEPS.edgeRealizationSum_right_eq_sum_edgeBlockedCoeff`, which rewrite the endpoint realization sums of `TNLean.PEPS.edgeInsertedCoeff_eq_sum_left_physicalRealization` and `TNLean.PEPS.edgeInsertedCoeff_eq_sum_right_physicalRealization` as weighted sums of the edge-blocked coefficient (using the endpoint update-invariance of the open middle weight, `TNLean.PEPS.edgeOpenMiddleWeight_update_left` and `TNLean.PEPS.edgeOpenMiddleWeight_update_right`); equality of PEPS states then transfers them between the two families (`TNLean.PEPS.edgeRealizationSum_left_sameState`, `TNLean.PEPS.edgeRealizationSum_right_sameState`), and `TNLean.PEPS.physical_to_virtual_insertion` supplies Y on the second family. The correspondence $X \mapsto Y$ is now realized as an *explicit* matrix-to-matrix map `TNLean.PEPS.edgeTransferMatrix`, built as a composition of algebra maps: insert X on the right endpoint of the first family and realize it (`TNLean.PEPS.edgeRightInsertionOp`), transfer the resulting physical operator to the second family across equal states, and read off the matrix it realizes there. The load-bearing identity that the explicit map agrees with the matrix recovered by `TNLean.PEPS.physical_to_virtual_insertion` is `TNLean.PEPS.edgeRightInsertionOp_realizes_edgeTransferMatrix`. Because each step preserves composition, the explicit map is multiplicative (`TNLean.PEPS.edgeTransferMatrix_mul`); it is also additive and homogeneous (`TNLean.PEPS.edgeTransferMatrix_add`, `TNLean.PEPS.edgeTransferMatrix_smul`) and satisfies the coefficient identity (`TNLean.PEPS.edgeTransferMatrix_edgeInsertedCoeff`). The multiplicativity that was *not* visible from the coefficient identity is thereby resolved.

   The full positive-bond theorem `TNLean.PEPS.isEdgeBlockedInsertionAlgebraIsomorphism` now proves that this correspondence is a $\mathbb{C}$-algebra

isomorphism, under the explicit hypotheses that every virtual bond of both tensor families has positive dimension. These hypotheses make explicit the nonzero virtual spaces used in Section 3, for example the reference residual configurations and the full matrix algebras on the virtual bonds. They are not consequences of injectivity. With a zero-dimensional bond, injectivity and state equality can hold vacuously while the physical-to-virtual recovery and gauge conclusion fail. The theorem-level obstruction is checked in `TNLean.PEPS.FundamentalTheoremCounterexample.fundamentalTheoremPEPSStatement_false`; the corresponding intermediate recovery failure is checked in `TNLean.PEPS.PhysicalToVirtualCounterexample`, and the edge-middle obstruction is `TNLean.PEPS.not_edgeMiddleTensorInjective_of_isEmpty`. Thus the positive-bond assumptions are the formal expression of the source's nonzero virtual bond spaces, not an unproved consequence. The last step in the positive-bond theorem is the injectivity of the edge-inserted coefficient in the inserted matrix: equal coefficients at every physical configuration determine the matrix. This gives the unit law $\Phi(1) = 1$ (the identity insertion has the same coefficient as $\Phi(1)$ by equal states), injectivity of the transfer map, and surjectivity by the symmetric $A \leftrightarrow B$ construction. The injectivity is the inverse-direction content of the resonate inversion used inside `TNLean.PEPS.physical_to_virtual_insertion`.

4. The algebra isomorphism is converted into an invertible edge gauge by the same finite-dimensional matrix-algebra argument as the one-dimensional injective theorem. This is formalized by `TNLean.PEPS.edgeGaugeFromInsertionAlgebraIsomorphism`: an algebra isomorphism between two full matrix algebras first identifies their sizes and is then conjugation by an invertible matrix. The per-edge gauges used in the global proof are collected by `TNLean.PEPS.exists_edgeGaugeFamily`.

5. After all edge gauges have been absorbed into the second tensor family, producing the modified tensors $\widetilde{B}_v$, the analog of `eq:inj_equal_edge` is formalized by `TNLean.PEPS.post_absorption_edge_insertion_equality`. Its conclusion constructs `TNLean.PEPS.PostAbsorptionEdgeInsertionEquality`: arbitrary insertions of the same matrix on the same edge give equal PEPS coefficients for the first tensor family and the absorbed second tensor family.

6. The generalized two-injective-tensor comparison from `lem:inj_equal_tensors_2` is formalized by `TNLean.PEPS.two_injective_tensor_insertion_comparison`. It says that equality of all virtual-bond insertions for two pairs of injective tensors forces $A_1 = \lambda B_1$ and $A_2 = \lambda^{-1} B_2$. The formal statement is an abstract version with nonempty spectator external boundary spaces; the source lemma is recovered by taking these spaces to be one-point spaces. The rank-one cancellation after coefficientwise product equality is now formalized by `TNLean.PEPS.twoBlockReciprocalScalarProportional_of_pointwise_mul_eq`: if $A_1(\eta_1, \mu, \sigma_1)A_2(\eta_2, \nu, \sigma_2) =$

$B_1(\eta_1, \mu, \sigma_1)B_2(\eta_2, \nu, \sigma_2)$ for all indices, then injectivity of A_1 and A_2 gives the reciprocal scalar proportionality. The one-shared-bond coefficient-separation case is now formalized by `TNLean.PEPS.two_injective_tensor_insertion_comparison_singletonBond`: inserting the matrix unit $E_{\mu,\nu}$ extracts the single product $A_1(\eta_1, \mu, \sigma_1)A_2(\eta_2, \nu, \sigma_2)$, and the pointwise product cancellation theorem then gives reciprocal scalar proportionality. The coefficientwise product separation above is on the source path only in the single-shared-bond case (`TNLean.PEPS.two_injective_tensor_insertion_comparison_singletonBond`), where a matrix-unit insertion extracts the product directly. In the many-bond case the source does not separate the product: it inverts A_2, reads off the gauges Z, U, W on three leg pairs, and uses their identity-form compatibility to force each gauge to be a scalar (`TNLean.PEPS.threeLeg_residual_forms_scalar`; `Papers/1804.04964/paper_normal.tex`, lines 1157–1204). The inversion and leg-regrouping are contained in the proved core comparison `TNLean.PEPS.two_injective_tensor_insertion_comparison_core`.

7. Both injective blocks of the one-vertex-versus-complement comparison are now formalized. The selected vertex block is `TNLean.PEPS.vertexTwoBlock` with injectivity `TNLean.PEPS.isTwoBlockInjective_vertexTwoBlock`. The complement block is `TNLean.PEPS.complementTwoBlock`, the contraction of A over $V \setminus \{v\}$ opened along the star $\mathsf{IncidentEdge}(G, v)$ as the shared boundary; its injectivity `TNLean.PEPS.isTwoBlockInjective_complementTwoBlock` follows from vertex injectivity and positive bond dimensions through `TNLean.PEPS.regionBlockedTensorInjective_of_isVertexInjective`, specialized to $V \setminus \{v\}$. The boundary-edge and physical-vertex identifications in `TNLean.PEPS.regionBlockedWeight_vertexComplement` identify this blocked tensor with the vertex-complement tensor. Thus `TNLean.PEPS.vertexComplementTensorInjective_of_isVertexInjective` is the same contraction-of-injective-tensors fact, rather than a second kernel-descent argument.

8. The final two-tensor comparison is applied by blocking one vertex against its complement. The paper uses this to prove $A_i = \lambda_i \widetilde{B}_i$ at each vertex, and then incorporates the scalars into the local gauges. This specialization is represented by `TNLean.PEPS.one_vertex_complement_comparison`. The formal statement uses the abstract two-block notation with nonempty external boundary spaces; the source comparison is obtained from the one-vertex and complement blocks after `eq:inj_equal_edge`.

9. The boundary-insertion argument removes the separate bond-dimension-identification hypothesis in `TNLean.PEPS.fundamentalTheorem_PEPS_of_bondDim`. For each edge, the algebra equivalence `TNLean.PEPS.edgeTransferAlgEquiv` identifies the two full matrix algebras of virtual insertions. The dimension count `TNLean.PEPS.bondDim_eq_of_matrixAlgEquiv` forces equality of the corresponding bond dimensions, and `TNLean.PEPS.fundamentalTheorem_PEPS` then gives the source gauge-equivalence

conclusion for the connected positive-bond statement. The connectivity hypothesis is the repair recorded in `docs/paper-gaps/peps_gaugeConsistency_connectivity_gap.tex`.

4 Source-to-formalization ledger

The following ledger records the present status of the Section 3 proof route. It is meant to prevent the proof from drifting away from the source argument.

- `eq:block_to_mps` edge blocking: `edgeBlockedCoeff_eq_stateCoeff` is locally proved; the transfer of injectivity is stated as the positive-bond theorem `TNLean.PEPS.IsVertexInjective.edgeBlockedThreeSiteInjective`. The middle physical index of this three-site object is formalized by `TNLean.PEPS.EdgeMiddlePhysicalConfig`. The named injectivity statement is `TNLean.PEPS.EdgeBlockedThreeSiteInjective`, and the endpoint assertions are proved from vertex injectivity, both for a single chosen edge and uniformly over all edges. The comparison from edge-middle region injectivity to edge-middle tensor injectivity is also available uniformly over all edges. The all-edge conditional implication from middle-region injectivity is `TNLean.PEPS.EdgeMiddleRegionInjectivityComparison.edgeBlockedThreeSiteInjective_all`.
- Identity insertion on the chosen bond: `thm:peps_edgeInsertedCoeff_identity` is formalized by finite diagonal reindexing.
- Lemma `inj_isomorph`, $X \mapsto O_1, O_2$: `thm:peps_virtualInsertionPhysicalRealization` formalizes the endpoint realization, and `thm:peps_edgeInsertedCoeff_endpointPhysicalRealization` formalizes the endpoint coefficient expansion.
- Lemma `inj_isomorph`, $O_1, O_2 \mapsto W$: `TNLean.PEPS.physical_to_virtual_insertion` proves the source recovery step. The middle block is contracted out by `TNLean.PEPS.resonate_middle_inverted`; the two endpoints are inverted by `TNLean.PEPS.resonate_invert_right_endpoint` and `TNLean.PEPS.resonate_invert_left_endpoint`; the two extracted matrices are reconciled by `TNLean.PEPS.resonate_endpoint_coeff_reconcile`. The positivity hypothesis on the virtual bond dimensions supplies the reference residual configurations used to fix the recovered matrix. The endpoint projected-recovery component for a common bond matrix is also formalized by `TNLean.PEPS.edgeEndpointLocalVirtualOpOfPhysicalOp_eq_of_projected_realization_eq`, and the conditional theorem `TNLean.PEPS.edgePhysicalToVirtualInsertion_of_projected_realization_eq` records the endpoint conclusion under explicit image-preservation hypotheses.
- Matrix-insertion algebra isomorphism, positive-bond variant: `TNLean.PEPS.isEdgeBlockedInsertionAlgebraIsomorphism` states the edge-blocked matrix-algebra correspondence and its inserted-coefficient equality

under positive bond dimensions for both tensor families. The inserted-coefficient correspondence $X \mapsto Y$ is formalized by `TNLean.PEPS.exists_edgeInsertedCoeff_eq`; its witness is the explicit transfer matrix and its coefficient identity is `TNLean.PEPS.edgeTransferMatrix_edgeInsertedCoeff`. The algebra-equivalence theorem `TNLean.PEPS.isEdgeBlockedInsertionAlgebraIsomorphism` is now proved in this positive-bond scope: well-definedness, injectivity, surjectivity, and the homomorphism laws including multiplicativity are part of the formalized statement. The positive-bond hypotheses express the source's nonzero virtual bond spaces; they are not consequences of injectivity.

- Edge gauge from the algebra isomorphism: `TNLean.PEPS.edgeGaugeFromInsertionAlgebraIsomorphism` proves the Skolem–Noether step from an algebra equivalence of full matrix algebras to conjugation by an invertible edge matrix. The edgewise family used later is `TNLean.PEPS.exists_edgeGaugeFamily`.

- Absorption of edge gauges into B: `TNLean.PEPS.absorbEdgeGauges` defines the absorbed tensor family, and `TNLean.PEPS.post_absorption_edge_insertion_equality` applies the edge-gauge family to it.

- Equation `eq:inj_equal_edge`: `TNLean.PEPS.post_absorption_edge_insertion_equality` constructs `TNLean.PEPS.PostAbsorptionEdgeInsertionEquality`.

- Lemma `inj_equal_tensors_2`, scalar step: the rank-one product cancellation is `TNLean.PEPS.twoBlockReciprocalScalarProportional_of_pointwise_mul_eq`, and the many-bond scalar forcing is `TNLean.PEPS.threeLeg_residual_forms_scalar`.

- Lemma `inj_equal_tensors_2`, comparison: `TNLean.PEPS.two_injective_tensor_insertion_comparison` states the abstract two-block insertion comparison with nonempty external boundary spaces and reciprocal scalar proportionality. The source lemma is the one-point external-boundary specialization.

- One vertex versus its complement: `TNLean.PEPS.one_vertex_complement_comparison` records the application of the generalized two-block comparison to a selected vertex and its complement, under the abstract nonempty external-boundary formulation.

- Final gauge equivalence and uniqueness: `TNLean.PEPS.fundamentalTheorem_PEPS_of_bondDim` proves the gauge-equivalence theorem after the bond dimensions are identified; `TNLean.PEPS.fundamentalTheorem_PEPS` removes that separate hypothesis by the matrix-algebra dimension count, and `TNLean.PEPS.gauge_unique_mod_edge_scalars` proves uniqueness modulo edge scalars.

5 Blueprint diagrams

The geometric steps are drawn directly where they enter the argument, while steps that are intrinsically algebraic are stated by their component equations.

- The inline edge-blocking reduction depicts the chosen edge and the passage to the three-site chain. Its endpoint labels follow `eq:block_to_mps`: the two selected endpoint regions are A'_1 and A'_2, while the complement is A'_3.
- The comparison of a virtual insertion in the two closed three-site chains is drawn inline in the statement of `lem:inj_isomorph`.
- The realization of a virtual insertion as a physical operation on either neighboring site is drawn as the corresponding three-panel equality.
- The converse implication, in which two neighboring physical operations determine one virtual operation on their shared bond, is also drawn inline.
- After the edge gauges have been absorbed into the second tensor family, the five-site comparison with its distinguished $(2, 5)$-edge insertion is drawn inline beside `eq:inj_equal_edge`.
- The comparison of two injective tensors with a matrix inserted on a shared virtual bond is drawn inline in the three-shared-bond subcase, with the external virtual indices shown explicitly.
- The four residual-operator forms involving Z, U, and W are drawn inline in the proof of `lem:inj_equal_tensors_2`.
- The inserted-edge coefficient is given by its defining contraction. Once the edge, physical indices, and inserted matrix are fixed, this coefficient is a scalar; a three-open-leg picture would describe a different object.
- The orientation convention for an edge gauge is stated by the two endpoint actions, rather than by a separate contraction.
- The action of all incident edge gauges on a vertex is stated by its valence-independent component equation.
- Cancellation of the two endpoint actions on an internal edge is stated as the corresponding matrix-inverse identity.
- The local gauge obtained from the blocked-middle comparison is stated by its component equation at an arbitrary vertex.
- Absorption of the incident edge gauges into B_v is stated by the vertexwise equation defining $\widetilde{B}_v$.
- Extraction of the local gauge is stated by the component equation that expresses B_v as the action of the incident edge matrices on A_v.

- Global consistency is stated by the single gauge family satisfying the local component equation at every vertex.

The inline comparison $A_R A_v = A_S \propto \widetilde{B}_S = \widetilde{B}_R \widetilde{B}_v$ belongs to the normal-region argument of [MGRPG$^+$18, Section 4], where $R \subset S = R \cup \{v\}$; it is not part of the one-vertex-versus-complement theorem in Section 3.

The retained and inline diagrams keep the local dot convention used in the blueprint, but preserve the distinctions that matter in the paper: tensor sites, blocked regions, virtual matrix insertions, and physical operations are not drawn as the same object. The component equations are used precisely when a separate picture would obscure the boundary data or assert a contraction not present in the source.

6 Conclusion

The connected positive-bond version of the injective PEPS Fundamental Theorem from [MGRPG$^+$18, Section 3] is now formalized. The existence clause `TNLean.PEPS.fundamentalTheorem_PEPS` and the uniqueness clause `TNLean.PEPS.gauge_unique_mod_edge_scalars` are proved and axiom-clean. The three-site injectivity transfer, the matrix-insertion comparison as a $\mathbb{C}$-algebra isomorphism, the edge gauge construction, and the one-vertex-versus-complement comparison are all complete. The positive-bond-dimension hypothesis records the source's nonzero virtual bond spaces; dropping it is refuted by `TNLean.PEPS.FundamentalTheoremCounterexample.fundamentalTheoremPEPSStatement_false`. Connectivity is an explicit repair of the literal source statement, documented in `docs/paper-gaps/peps_gaugeConsistency_connectivity_gap.tex`; dropping it is refuted by `TNLean.PEPS.GaugeConsistencyConnectivityCounterexample`.

References

[MGRPG$^+$18] András Molnár, José Garre-Rubio, David Pérez-García, Norbert Schuch, and J. Ignacio Cirac. Normal projected entangled pair states generating the same state. *New Journal of Physics*, 20(11):113017, 2018.