Normal PEPS Fundamental Theorem: Section 3 Route

2026-05-06

Verdict: scope-restriction (open).

1 The source argument

The normal-PEPS part of [MGRPG+18] begins after the injective PEPS theorem. Its purpose is not to replace the injective proof, but to make the injective proof applicable after a controlled blocking. A normal PEPS is one for which sufficiently large regions can be blocked into injective tensors. Once such injective regions are available, the proof again applies the three-site injective-chain argument around a chosen edge.

The source proof has three layers. First, Lemma `injective_union` proves that the union of two injective regions is injective. The proof blocks the network into the four regions $A \setminus B$, $A \cap B$, $B \setminus A$, and $(A \cup B)^c$. If a boundary tensor kills the contraction over $A \cup B$, injectivity of A first removes the $A \setminus B$ and $A \cap B$ part, and injectivity of B then removes the remaining $A \cap B$ and $B \setminus A$ part. [1]

Second, for the translationally invariant square-lattice theorem, the paper assumes that every 2×3 and 3×2 region is injective. By repeatedly applying the union lemma, the regions denoted R, S, and T in the paper are injective. The finite-lattice block denoted A_3 in the proof of Theorem `3` has the same complementary shape as T but lives in the full finite lattice around the chosen edge; the present coordinate definition records only the displayed local-window complement. The regions R and S differ by one site, and their complements are injective when the system is large enough. These are the regions used in the last comparison step. [2]

Third, the proof of Theorem `3` blocks a neighborhood of each edge into three injective pieces. The red region is A_1, the blue region is A_2, and the complement is A_3. This produces the same three-site injective MPS configuration used in the injective PEPS proof. Lemma `inj_isomorph` therefore gives a gauge on

[1] The source is `Papers/1804.04964/paper_normal.tex:1324--1400`.

[2] The regions R, S, and T are drawn at `Papers/1804.04964/paper_normal.tex:1407--1443`.

every edge. Translation invariance reduces these edge gauges to one horizontal matrix X and one vertical matrix Y. After absorbing them into the second tensor, the final comparison of the R and S regions gives $A \propto \widetilde{B}$. [3]

The general theorem `normal` removes the square-lattice specialization. It assumes precisely the blockings needed above: every edge can be blocked into a three-site injective chain, and every site admits two injective regions with injective complements that differ exactly in that site. [4]

2 Current formal representation

The injective PEPS objects used by the normal theorem are represented in `TNLean/PEPS/Defs.lean`, `TNLean/PEPS/VirtualInsertion.lean`, and `TNLean/PEPS/Blocking.lean`. The PEPS chapter of the blueprint has a separate normal-PEPS section after the injective theorem. This separation is important: the normal theorem needs extra blocking hypotheses before the injective proof route can be invoked.

The present blueprint nodes are the following.

- `lem:peps_injectiveRegion_union` states the union-of-injective regions lemma. It is tracked by GitHub issue #1374. It records the four-region decomposition and the two successive inverse applications used in the source proof.
- `lem:peps_regionInjectivityUnionClosure_finite` records the finite nonempty iteration of the binary union-closure assertion.
- `def:peps_normalSquareBlockingRegions` records the square-lattice regions R, S, and T. It is part of GitHub issue #1375. The coordinate rectangular hypotheses, union closure, and a supplied rectangular cover of the displayed T-region imply this abstract structure.
- `thm:peps_normalEdgeBlockingToInjectiveChain` states that the normal square-lattice hypotheses allow a blocking around every edge into three injective regions. It is part of GitHub issue #1375.
- `thm:peps_tiNormalGaugeAbsorption` records the translationally invariant gauge absorption with horizontal gauge X and vertical gauge Y. It is part of GitHub issue #1375.
- `thm:fundamentalTheorem_normalSquarePEPS` records the square-lattice theorem corresponding to Theorem 3. It is part of GitHub issue #1375.
- `thm:fundamentalTheorem_normalPEPS` records the general normal PEPS theorem following Theorem 3. It is part of GitHub issue #1375.

[3] The proof of Theorem 3 is `Papers/1804.04964/paper_normal.tex:1449--1572`.
[4] The general theorem is `Papers/1804.04964/paper_normal.tex:1574--1585`.

The finite rectangular square lattice is represented by pairs of horizontal and vertical coordinates. Its graph has horizontal and vertical nearest-neighbor edges, named coordinate right/up edges, and orientation predicates distinguishing the two possible directions. Every edge has exactly one of these orientations. With the ordered endpoints convention for edges, horizontal edges are coordinate right edges from their left endpoints, and vertical edges are coordinate upward edges from their lower endpoints.

The four-region decomposition from Lemma `injective_union` is formalized both by named regions and by an indexed family in the source-paper order: $A \setminus B$, $A \cap B$, $B \setminus A$, and $(A \cup B)^c$. The indexed family reconstructs A, B, $A \cup B$, A^c, B^c, and the full vertex set. The same reconstructions are available as finite unions over the selected indexed subfamilies $\{0, 1\}$, $\{1, 2\}$, $\{0, 1, 2\}$, $\{2, 3\}$, and $\{0, 3\}$. The singleton subfamily $\{3\}$ gives the outside block and is the complement of the selected inside subfamily $\{0, 1, 2\}$. Its selected complementary pairs match the two places where the source proof applies injectivity of A and then injectivity of B. The selected subfamilies $\{0, 1\}$ and $\{2, 3\}$ are disjoint, as are $\{1, 2\}$ and $\{0, 3\}$; these are the disjointness forms needed for the two inverse applications. The abstract union-closure hypothesis also has the corresponding indexed inside-union form: injectivity of the selected subfamilies $\{0, 1\}$ and $\{1, 2\}$ implies injectivity of the inside subfamily $\{0, 1, 2\}$. Every vertex belongs to exactly one indexed piece. These facts prepare the tensor-level boundary-contraction proof, but do not prove that theorem.

The rectangular vocabulary separates coordinate products from source-paper contiguous blocks. Coordinate rectangles are products of finite coordinate sets, whereas contiguous coordinate rectangles are products of coordinate intervals. The 2×3 and 3×2 coordinate-product predicates assert only the two cardinalities; the contiguous-rectangle predicates express the source-paper blocks. The square-lattice rectangular injectivity hypotheses use these two contiguous predicates and map to the abstract rectangular-injectivity bundle. Bounded contiguous rectangles of sizes 2×3 and 3×2 satisfy the corresponding source-rectangle predicates.

The displayed regions R and S are explicit unions of contiguous rectangles. Region R is the union of one contiguous 2×3 rectangle and one contiguous 3×2 rectangle. Region S is the union of two contiguous 2×3 rectangles. Their coordinate definitions agree with the source claim that they differ in one site: $R \subseteq S$, and $S \setminus R$ is the lower-right site of the displayed 3×3 square. Assuming the union-closure assertion supplied by the source lemma on unions of injective regions, these decompositions imply injectivity of R and S. The formal development also contains the finite nonempty iteration of this union-closure assertion, matching the repeated unions used after Lemma `injective_union`.

The displayed local model for T is the complement, inside the source 5×6 window, of the two edge blocks in the source picture. These removed blocks are one contiguous 2×3 rectangle and one contiguous 3×2 rectangle, hence they are injective under the coordinate rectangular hypotheses. They are disjoint, and together with the local-window T region they reconstruct the 5×6 window. This local-window region is distinct from the finite-lattice complementary block A_3

used later in the proof of Theorem 3. The finite-lattice block is the complement of the two edge blocks in the full square lattice; its elementary disjointness and covering identities are formalized separately.

A target-parametric rectangular-cover criterion supplies a sufficient injectivity route: if a square-lattice region has a nonempty cover by contiguous 2×3 and 3×2 rectangles, then rectangular injectivity and finite union closure imply injectivity of that region. The formal special cases are the local-window T region and the finite-lattice complementary block. This criterion is only a sufficient local model, not the source argument itself.

The origin-based local-window model for T has no such rectangular cover whenever the ambient lattice contains the corresponding 5×6 window. The obstruction is a point-forcing argument. For the local-window target T_{loc} at the origin, let $R^{a\times b}_{x,y}$ denote the contiguous $a \times b$ rectangle with lower-left corner (x, y). Then

$$p := (0,0) \in T_{\mathrm{loc}}, \tag{1}$$

$$p \in R^{2\times 3}_{x,y} \implies q_{23} := (0,2) \in R^{2\times 3}_{x,y}, \qquad q_{23} \notin T_{\mathrm{loc}}, \tag{2}$$

$$p \in R^{3\times 2}_{x,y} \implies q_{32} := (2,1) \in R^{3\times 2}_{x,y}, \qquad q_{32} \notin T_{\mathrm{loc}}. \tag{3}$$

Thus any exact cover of T_{loc} by 2×3 and 3×2 rectangles would cover p by one of its pieces and would therefore also cover a point outside T_{loc}. The same calculation applies to the rotated local-window model for vertical edges in a 6×5 window, to the origin-based horizontal edge-complement model in the 5×7 frame, and to the origin-based vertical edge-complement model in the 7×5 frame. The forced points are

target	p	q_{23}	q_{32}
T_{loc}	$(0,0)$	$(0,2)$	$(2,1)$
$T^{\mathrm{rot}}_{\mathrm{loc}}$	$(0,0)$	$(1,2)$	$(2,1)$
$A_{3,\mathrm{hor}}$	$(0,0)$	$(0,2)$	$(2,1)$
$A_{3,\mathrm{ver}}$	$(0,0)$	$(1,2)$	$(2,1)$

The formal point-forcing lemma is `TNLean.PEPS.not_squareLatticeRectangleCover_of_forced_points`. Its four present applications are the corresponding no-cover theorems for the local T models and the edge-complement models. These no-cover results do not contradict the paper. They show that the source's injectivity argument for T and A_3 must be aligned with the finite PEPS geometry, rather than read directly from the present exact-cover criterion.

The normalized horizontal edge frame has a collar identity: in the 5×7 frame, A_3 is the local-window T region together with two top-collar contiguous 3×2 rectangles. Thus local T-injectivity implies injectivity of A_3 by union closure and rectangular injectivity. The vertical counterpart is a 7×5 frame in which the edge complement is the rotated local T region together with two right-collar contiguous 2×3 rectangles. A rectangular cover of the rotated local T region therefore implies injectivity of the vertical complementary block.

The injectivity of A_3 is now established directly, with no cover hypothesis and no detour through the local-window T region. The local T region is a fixed small window whose own corner is forced, so it has no rectangular cover by contained source rectangles; but A_3 is the complement of the removed L-shape in the *full* finite lattice, and when the removed shape sits in the interior the corner has room. The realignment is a finite-geometry cover of A_3 by four full-length bands around the removed shape — below, above, left, and right — together with two filler rectangles inside the shape's bounding box. Each band is a contiguous rectangle whose two side lengths are at least the source 2×3 or 3×2 shapes, hence injective as a finite union of those shapes; each filler is one source rectangle. The union of these six rectangles equals the complement of the removed shape, so union closure and rectangular injectivity give injectivity of A_3 unconditionally for an interior edge.[5] The interior placement asks for one row and column of margin below and to the left of the removed shape and enough total room above and to the right, so the shape is surrounded on every side.

The normalized horizontal and vertical edge blockings give one-edge data with red, blue, and complementary regions. The distinguished edges have the expected orientations; their endpoints lie in the red and blue regions; the three regions are pairwise disjoint and cover the finite lattice. This one-edge datum is the local form of the later every-edge blocking hypotheses. A family of such one-edge data determines those hypotheses, and each datum supplies the three injective regions used in the associated edge-blocked chain.

The translated horizontal and vertical pictures carry the same data at an arbitrary coordinate origin. When the translated frame fits in the finite lattice, the translated edge is identified with the corresponding coordinate right or upward edge. A rectangular cover of the translated complementary region supplies the one-edge blocking datum and hence the three injective regions. The normalized pictures are the origin specializations of these translated pictures, while the normalized and translated data remain in separate Lean modules.

The conditional every-edge construction has two equivalent forms. A translated edge-window assignment gives, for every edge, a horizontal or vertical window and a rectangular cover of its complement. Such an assignment implies the normal edge-blocking hypotheses. An oriented margin-and-cover datum records the same input directly on each square-lattice edge: the edge orientation, the coordinate inequalities placing the translated frame, and the complementary cover. A family of these data also determines the normal edge-blocking hypotheses, and the resulting hypotheses recover the supplied one-edge datum at each edge.

With the interior A_3 injectivity in hand, the every-edge construction is now cover-free at interior edges. For an edge with interior margins — horizontal with enough room to place the surrounded horizontal removed shape, or vertical with enough room for the rotated one — the translated edge-blocking datum

[5] Formal declarations: `TNLean.PEPS.NormalSquareLatticeRectangleInjectivityHypotheses.wideRectangle_injective`, `TNLean.PEPS.NormalSquareLatticeRectangleInjectivityHypotheses.shortRectangle_injective`, `TNLean.PEPS.NormalSquareLatticeRectangleInjectivityHypotheses.interiorEdgeComplement_injective`, and `TNLean.PEPS.NormalSquareLatticeRectangleInjectivityHypotheses.interiorVerticalEdgeComplement_injective`.

is built with no rectangular cover: its complementary-block injectivity is the interior A_3 injectivity above. A family of these cover-free interior data over all edges assembles the normal edge-blocking hypotheses, with the injective chain, endpoint membership, pairwise disjointness, and coverage available at every edge.[6]

For the present open rectangular coordinate graph, the margin predicates describe only interior translated frames. A coordinate right edge must have enough left and right margin; a coordinate upward edge must have enough bottom and top margin. Hence the cover-free interior construction supplies data only on the interior edges of the open 7×7 graph; the earlier translated-window and oriented margin-and-cover criteria have the same interior restriction. Explicit boundary obstructions are recorded on all four sides. The remaining every-edge construction must therefore include the boundary geometry: the interior A_3 injectivity removes the local and rotated T-cover obligation for interior edges, leaving the boundary edges of the open rectangular model.

Verdict: this note records an open gap with a scope restriction. The earlier vocabulary gap for contiguous rectangular blocks is closed. The earlier coordinate rectangular-injectivity structure is also present, and the indexed finite-region geometry of Lemma `injective_union` is formalized down to the selected complement and closure forms used by the two inverse applications. The rectangular decompositions of R and S are formalized, and the conditional derivations of injectivity for R and S from union closure are also formalized. The two edge blocks removed from the displayed T-window are injective under the coordinate rectangular hypotheses. The finite-lattice complementary block around the chosen edge has a set-theoretic definition. The exact-cover criterion by 2×3 and 3×2 rectangles is only a restricted sufficient condition, and the origin-based instances above show that it is not the source's injectivity argument. Injectivity of the edge-complementary block A_3 is now derived unconditionally for an interior edge, by the finite-geometry band-and-filler cover of the full-lattice complement; the local-window T region is no longer on the critical path, since the source object is A_3 and the local window is a fixed small frame whose corner is forced. The every-edge blocking construction is cover-free at interior edges and assembles the normal edge-blocking hypotheses there. The tensor-level union theorem is proved for positive bond dimensions, including the overlapping case (the overlap re-insertion). The remaining open gap is the boundary geometry of the open rectangular model: edges near the lattice boundary lack interior margins, so the cover-free construction does not yet reach them; on a translation-invariant lattice every edge is interior and the construction is complete. Injectivity of R and S is already derived from union closure and rectangular injectivity.

The normal route itself is tracked by GitHub issue #1373. The normal-section diagram work is tracked by GitHub issue #1376. The general PEPS status tracker is GitHub issue #1252.

[6]Formal declarations: `TNLean.PEPS.horizontalSquareLatticeEdge_blockingDatum_interior`, `TNLean.PEPS.verticalSquareLatticeEdge_blockingDatum_interior`, `TNLean.PEPS.NormalSquareInteriorEdgeDatum`, and `TNLean.PEPS.normalSquareEdgeBlockingHypotheses_of_interiorData`.

3 Remaining mathematical obligations

The normal PEPS theorem cannot be obtained by citing the injective theorem alone. The lists below are the authoritative record for the square-lattice normal blocking route. Lean docstrings and blueprint entries should point here rather than repeat this status in full.

The following part of the formalization is already present.

- Region injectivity is represented by `TNLean.PEPS.RegionInjectivityData`; the same infrastructure contains the region-complement definitions `TNLean.PEPS.regionComplement` and `TNLean.PEPS.regionOutsideUnion`, and the finite-region union-closure hypothesis `TNLean.PEPS.RegionInjectivityUnionClosure`.
- The four-region decomposition for Lemma `injective_union` is formalized both by named regions and by the indexed family $A \setminus B$, $A \cap B$, $B \setminus A$, and $(A \cup B)^c$, including the reconstruction, disjointness, and selected-complement identities used by the two inverse applications.
- The three contraction maps for Lemma `injective_union` are separated as `TNLean.PEPS.InjectiveUnionContractionMaps`.
- The abstract contraction-chain criterion for Lemma `injective_union` is formalized by `TNLean.PEPS.InjectiveUnionContractionChain`: if the three implications

$$\mathcal{C}_{\{0,1,2\}}(X) = 0 \Longrightarrow \mathcal{C}_{\{2\}}(X) = 0 \Longrightarrow \mathcal{C}_{\{1,2\}}(X) = 0 \Longrightarrow X = 0$$

  are supplied, then the union contraction has trivial kernel.
- The square-lattice coordinate model, contiguous 2×3 and 3×2 rectangles, and the coordinate rectangular-injectivity hypotheses are present.
- The source regions R and S are explicit finite coordinate regions, their rectangular decompositions are formalized, and their injectivity follows conditionally from rectangular injectivity and union closure.
- The displayed local T-region, the finite-lattice edge-complement regions, their sufficient rectangular-cover criteria, the no-cover diagnostics for the present origin-based local models, and the normalized collar decompositions are formalized.
- The one-edge, translated-window, and margin-cover hypotheses for normal edge blocking are present, together with their endpoint, disjointness, coverage, and injective-chain consequences.
- The general normal PEPS blocking hypotheses `TNLean.PEPS.NormalPEPSBlockingHypotheses` provide their edge and one-site consequences directly, so downstream square-lattice arguments can use these hypotheses rather than restating the edge-blocking and one-site hypotheses separately.

The remaining obligations are the following.

1. The tensor-level union-of-injective-regions lemma is now proved, with the explicit positive-bond hypothesis the formalization carries. The proof follows the source by applying the left inverses for A and B in sequence. The disjoint case is formalized over a bare three-block geometry (no red-block injectivity and no distinguished edge): for disjoint injective regions S and T with positive bond dimensions, $S \cup T$ is injective, and its nonempty finite iteration over pairwise-disjoint families is also available.[7] The general overlapping statement $\mathrm{inj}(A) \wedge \mathrm{inj}(B) \Rightarrow \mathrm{inj}(A \cup B)$ is now closed for positive bond dimensions.[8] The displayed regions R and S are unions of *overlapping* rectangles, so they require this four-region two-inverse descent rather than the disjoint engine.

   The closure (the overlap re-insertion). The remaining obstruction recorded below was the P_0-outer freedom of the right rebuild: the right-rebuild row is indexed by R_2 boundary configurations, and a row over R_2 alone cannot separate the P_0-outer host indices. The closure carries those indices through the re-insertion as the source's open legs of the final tensor. The union boundary edges partition into the R_2 boundary edges and the P_0-outer edges (the union boundary edges running from $R_1 \setminus R_2$ to $(R_1 \cup R_2)^c$), and the P_0-outer edges are disjoint from the R_2, overlap, and difference boundary edges while lying on the boundary of R_1. Fixing the P_0-outer label to a reference δ and restricting the coefficient family to the P_0-fiber of δ gives a row whose right coupling, read through the overlap-crossing collapse and the bridge coefficient identity, is a sum of the left first strips restricted to the same fiber. Each restricted strip is either zero (the R_1 boundary label determines the P_0-outer label, so a mismatched fiber kills it) or the full strip (which already vanishes). The right rebuild then forces the fiber row to vanish after inverting R_2, and the partition determinacy selects a single host term: fixing δ to the P_0-outer part and b_2 to the R_2 part of a configuration realizing a host label forces the coefficient to vanish there. Union host-boundary surjectivity covers every label.[9] The analysis below records how this closure was reached and is retained as the development history.

   The two host three-block geometries of the overlapping descent are now formalized over a bare geometry, together with the first inverse application. With the four parts $P_0 = A \setminus B$, $P_1 = A \cap B$, $P_2 = B \setminus A$, and $P_3 = (A \cup B)^c$, the host of both geometries is $A \cup B = P_0 \cup P_1 \cup P_2$ and their shared inner

[7] Formal declarations: `TNLean.PEPS.regionBlockedTensorInjective_union_disjoint` and `TNLean.PEPS.regionBlockedTensorInjective_biUnion_disjoint`.

[8] Formal declaration: `TNLean.PEPS.regionBlockedTensorInjective_union_overlap`, with `TNLean.PEPS.regionBlockedTensorInjective_union_disjoint` the disjoint specialization.

[9] Formal declarations: `TNLean.PEPS.p0OuterLabel`, `TNLean.PEPS.regionBoundaryLabel_union_eq_of_R_p0Outer`, `TNLean.PEPS.overlapLeft_firstStrip_fiber_weightCombination_eq_zero`, `TNLean.PEPS.overlapFiber_bridge_rightCoupling_eq_zero`, `TNLean.PEPS.overlapFiber_bridgeRow_eq_zero`, and `TNLean.PEPS.regionBlockedTensorInjective_union_overlap`.

block is the residual coupling block $P_2 = B \setminus A$. The left geometry places A as the blue block and P_2 as the complement block; the right geometry places the overlap P_1 as the blue block and P_2 as the complement block over the smaller host B. The first inverse application then strips A from the host annihilation and leaves the residual coupling through P_2 vanishing for every P_2 physical leg and every A-boundary configuration.[10]

The remaining sub-step is the re-insertion bridge of the source's "plug back the tensor over $A \cap B$" move, which closes through the right geometry over the smaller host B. The two residual couplings share the P_2 block but constrain different edge sets. The left geometry's coupling leaves the internal P_1–P_0 edges free, since both endpoints of such an edge lie in A and the edge is neither an $A \cup B$-boundary edge nor an A-boundary edge. The right geometry's B-host constraint pins exactly these edges, because a P_1–P_0 edge crosses the boundary of B ($P_1 \subseteq B$, $P_0 \cap B = \emptyset$). The bridge is therefore the multiplicity collapse over the P_1–P_0 crossing edges: contracting the overlap tensor along its boundary with $A \setminus B$ rebuilds the B-block weight from the left geometry's P_2 coupling, summed over the free P_1–P_0 indices. This is the larger-host analogue of the landed complement collapse `TNLean.PEPS.ThreeBlockGeometry.regionBlockedWeight_complement_eq_smul_constrained` and `TNLean.PEPS.ThreeBlockGeometry.blueRedCrossingBondProd_smul_threeBlockComplCoeff_eq`, with the crossing-edge family taken to be the P_1–P_0 edges and the produced block taken to be B. After the collapse, the right geometry's blue-side factorization `TNLean.PEPS.ThreeBlockGeometry.regionInteriorBondProd_smul_threeBlockBlueWeight_eq` reads the rebuilt coupling as a combination of B blocked weights, the left inverse for B forces the reconstructed host residual coefficients to vanish, and host-boundary surjectivity (`TNLean.PEPS.ThreeBlockGeometry.exists_regionBoundaryLabel_host_eq`) makes every host residual realized, closing $c = 0$. The new content is the P_1–P_0 crossing collapse; the rest reuses the landed three-block factorizations and the first inverse application `TNLean.PEPS.overlap_firstStrip`.

The right-geometry blue-side rebuild step is now landed: `TNLean.PEPS.overlapRight_bondProd_smul_hostWeight_combination_eq_zero` takes a coefficient row over the B-host boundary configurations whose right-coupling combination $\sum_{b_2} \mathrm{row}(b_2) \cdot \mathrm{cc}_R(b_2, \sigma_{P_2}, \beta_{P_1})$ vanishes for every difference leg σ_{P_2} and overlap boundary β_{P_1}, and produces a vanishing B-host-weight combination read through the fused overlap/difference leg, by the blue-side factorization `TNLean.PEPS.regionInteriorBondProd_smul_threeBlockBlueWeight_eq`. The P_1–P_0 crossing edges are also characterized (`TNLean.PEPS.IsOverlapCrossingEdge`): both endpoints in R_1, a $B = R_2$ boundary edge but neither an $A = R_1$ nor an

[10] Formal declarations: `TNLean.PEPS.overlapLeftGeometry`, `TNLean.PEPS.overlapRightGeometry`, and `TNLean.PEPS.overlap_firstStrip`, all built on the bare three-block factorization machinery of `TNLean.PEPS.ThreeBlockGeometry`.

$A \cup B$ boundary edge, and a blue/red crossing edge of the right geometry (`TNLean.PEPS.isBlueRedCrossingEdge_right_of_overlapCrossing`, using $P_0 = R_1 \setminus R_2 \subseteq R_2^c$).

The P_0-outer bridge is now landed. The P_0-outer edges (the $A \cup B$ boundary edges that are not B boundary edges, running $P_0 \to (A \cup B)^c$) are formalized (`TNLean.PEPS.IsP0OuterEdge`) together with the gluing along them (`TNLean.PEPS.p0OuterGlue`) and along the P_1–P_0 crossing edges (`TNLean.PEPS.interCrossGlue`). The two host residuals are reconciled by two gathering identities (`TNLean.PEPS.sum_hostGlue_mul_rightIndicator_eq` and `TNLean.PEPS.sum_interGlue_mul_leftIndicator_eq`), each proved by uniqueness of the realizing B- respectively A-boundary label (the determinacy lemmas `TNLean.PEPS.regionBoundaryLabel_R_eq_of_inter_sdiff` and `TNLean.PEPS.regionBoundaryLabel_R_eq_of_union_sdiff`) plus the gluing; both couplings reduce, by the landed crossing collapse, to the same P_2 blocked-region weights with an existence-indicator coefficient (`TNLean.PEPS.ThreeBlockGeometry.crossingBond_smul_complCoeff_combination_eq`). The scalar heart of the bridge is `TNLean.PEPS.overlapBridge_coeff_eq`. With the boundary-configuration transport between the geometry-native host univ $\setminus g$.red and the literal union regions (`TNLean.PEPS.regionBoundaryConfigCongr`), the first strip read through the crossing collapse vanishes (`TNLean.PEPS.overlapLeft_firstStrip_weightCombination_eq_zero`), and the bridge produces the rebuild hypothesis: `TNLean.PEPS.overlap_bridge_rightCoupling_eq_zero` discharges $\sum_{b_2} \mathrm{row}(b_2) \cdot \mathrm{cc}_R(b_2, \sigma_{P_2}, \beta_{P_1}) = 0$ for the bridge row `TNLean.PEPS.overlapBridgeRow`.

Feeding the summed bridge row to the rebuild and inverting B exposes one residual freedom: a row over the B-boundary alone does not separate the P_0-outer host indices. The live closure restricts the coefficient family to each P_0-outer fiber before applying the left strip and bridge. The resulting fiber row vanishes after the right rebuild, and the pair consisting of its B-boundary label and its P_0-outer label determines the union boundary label. This gives the final injectivity statement `TNLean.PEPS.regionBlockedTensorInjective_union_overlap`.

A full A-boundary parameter is unsuitable because a row indexed only by B-boundary configurations cannot see the overlap-to-difference edges inside B. A host-spanning three-block geometry also does not close the argument: it reduces to blocked weights of $A \setminus B$, while the hypotheses do not make that strict subregion injective.

The proof instead uses the left and right three-block geometries and their two inverse applications, carrying only the P_0-outer part of the A-boundary to close the missing host indices.

Resolution. The obstruction above is that a row indexed by the full A-boundary over-determines the R_1 label on the overlap-to-difference edges, which lie inside B and are invisible to a B-boundary row. The closure instead parametrizes by the P_0-outer label alone, the part of the A-boundary on the union boundary edges running into $(A \cup B)^c$. These edges are disjoint from the R_2, overlap, and difference boundary edges, so the P_0-outer label is an independent parameter that the right rebuild does not touch; the gathering glue along the P_0-outer edges then combines the two halves cleanly. This is the route described in the closure paragraph at the head of this obligation.

2. Injectivity of the complementary block A_3 is now realigned with the finite PEPS geometry and proved unconditionally for an interior edge. The local-window T region is a fixed small window whose own corner is forced and so has no rectangular cover; the source object is instead A_3, the complement of the removed L-shape in the full finite lattice, which for an interior edge is the union of four full-length bands around the removed shape and two filler rectangles. Each band is a large contiguous rectangle, injective as a finite union of source 2×3 (respectively 3×2) shapes; the union of the six pieces equals A_3, so union closure proves it injective with no cover hypothesis. The same realignment holds for the rotated vertical complementary block. The earlier exact-cover criterion remains a useful sufficient condition, but it is no longer needed for interior edges.[11]

3. The every-edge finite-geometry construction is now cover-free at interior edges. Each interior horizontal or vertical edge gets a translated edge-blocking datum whose complementary-block injectivity is the interior A_3 injectivity, with no rectangular cover; a family of these cover-free interior data over all edges assembles the normal edge-blocking hypotheses, with the injective chain, endpoint membership, pairwise disjointness, and coverage at every edge.[12] The remaining task is the boundary geometry of the open rectangular model, where some edges lack interior margins; on a translation-invariant (periodic) lattice every edge is interior and the construction is complete.

4. Reuse the injective-chain insertion route after this blocking: GitHub issue #1367, GitHub issue #1368, GitHub issue #1363, GitHub issue #1364, GitHub issue #1361, and GitHub issue #1360. The formal material for this route separates into unconditional region-insertion algebra, the identity-insertion bridge to the closed state, and a conditional matrix-transfer datum.

[11] Formal declarations: `TNLean.PEPS.NormalSquareLatticeRectangleInjectivityHypotheses.wideRectangle_injective`, `TNLean.PEPS.NormalSquareLatticeRectangleInjectivityHypotheses.shortRectangle_injective`, `TNLean.PEPS.NormalSquareLatticeRectangleInjectivityHypotheses.interiorEdgeComplement_injective`, and `TNLean.PEPS.NormalSquareLatticeRectangleInjectivityHypotheses.interiorVerticalEdgeComplement_injective`.

[12] Formal declarations: `TNLean.PEPS.horizontalSquareLatticeEdge_blockingDatum_interior`, `TNLean.PEPS.verticalSquareLatticeEdge_blockingDatum_interior`, `TNLean.PEPS.NormalSquareInteriorEdgeDatum`, and `TNLean.PEPS.normalSquareEdgeBlockingHypotheses_of_interiorData`.

The region-inserted coefficient is linear in the inserted matrix and, under the same positive-bond restriction as in the edge case, determines that matrix from all region and complement physical configurations. These facts use linear independence of the blocked weight family of R and of the complement $V \setminus R$, and give the region analogue of the injectivity of the edge-inserted coefficient.[13] Once an explicit edge matrix transfer has matched region-inserted coefficients, is multiplicative, and preserves the unit, it gives a region insertion-algebra isomorphism and hence, by Skolem–Noether, a per-edge gauge matrix.[14]

The identity-insertion bridge is the following equation. Here $C_A(R, f, X; \sigma, \tau)$ denotes the region-inserted coefficient for the tensor A, the region R, the boundary edge f, the boundary matrix X, the region physical configuration σ, and the complement physical configuration τ. The symbol $D_A(e)$ denotes the bond dimension of the edge e, ∂R denotes the set of boundary edges of R, and $\langle \sigma, \tau \mid \Psi_A \rangle$ denotes the closed-state coefficient of the assembled physical configuration:

$$C_A(R, f, 1; \sigma, \tau) = \Big(\prod_{e \notin \partial R} D_A(e) \Big) \langle \sigma, \tau \mid \Psi_A \rangle \tag{4.1}$$

The proof decomposes the closed coefficient into region and complement contractions, rewrites identity insertion as a double sum over boundary-agreeing virtual configurations, and collapses that double sum by merging the two virtual configurations. The factor $\prod_{e \notin \partial R} D_A(e)$ is therefore part of the formula, not a normalization that can be discarded.[15] Under matched bond dimensions the corresponding interior bond products for A and B agree, so the same factor is available in the matched-coefficient transfer.[16]

For a second tensor B, let $T_f(X)$ denote the boundary matrix assigned to X on the edge f by the desired region insertion transfer. The formal candidate pulls the in-region endpoint operator of A back through B's vertex and reads it off through a fixed residual reference frame, in analogy with the edge transfer matrix. The available recovery theorem is conditional: it reduces the matched coefficient identity to two hypotheses, namely that the transferred endpoint operator preserves the image of B's local tensor map at the in-region vertex and that its virtual pullback through B has

[13] Formal declarations: `TNLean.PEPS.regionInsertedCoeff_add`, `TNLean.PEPS.regionInsertedCoeff_smul`, `TNLean.PEPS.regionInsertedCoeff_injective`, and `TNLean.PEPS.edgeInsertedCoeff_injective`.

[14] Formal declarations: `TNLean.PEPS.RegionInsertionTransfer`, `TNLean.PEPS.isRegionBlockedInsertionAlgebraIsomorphism_of_transfer`, and `TNLean.PEPS.exists_regionEdgeGauge_of_transfer`.

[15] Formal declarations: `TNLean.PEPS.prod_assembleRegion_split`, `TNLean.PEPS.regionInsertedCoeff_identity_eq_doubleSum`, `TNLean.PEPS.regionMerge`, `TNLean.PEPS.regionProd_eq_merge`, `TNLean.PEPS.complementProd_eq_merge`, `TNLean.PEPS.regionInteriorBondProd`, `TNLean.PEPS.stateCoeff_eq_regionComplement`, and `TNLean.PEPS.regionInsertedCoeff_one_eq_stateCoeff`.

[16] Formal declarations: `TNLean.PEPS.regionInteriorBondProd_congr` and `TNLean.PEPS.regionInsertedCoeff_transfer_of_realizes`.

the single-edge insertion form on f.[17] These are the region analogues of the two consequences extracted by the edge physical-to-virtual recovery.[18]

A useful image-preservation argument is available only in the vertex-injective regime. If B is vertex injective, then the state-vector coefficients at the in-region vertex span the local virtual coefficient space by row–column rank duality for the vertex-complement tensor family of B. Under $\mathrm{SameState}(A, B)$ this identifies the local tensor images of A and B, and the transferred endpoint operator preserves that image.[19] This does not discharge the normal-PEPS route of [MGRPG+18, Section 3]: the source theorem assumes injectivity after blocking, not single-vertex injectivity. The region-injective proof still requires the same spanning and image-preservation statement derived from blocked-region injectivity.

Even in the vertex-injective setting, the full realization is conditional on the remaining matrix-structure transfer: the virtual pullback of the transferred endpoint operator must be the matrix insertion of $T_f(X)^\mathsf{T}$ on f.[20] Equivalently, the change of frame between A's and B's coefficient spaces at the in-region vertex must intertwine the single-bond insertions on f. This is the region analogue of the incident-matrix form read from the edge resonate identity.

This matrix-structure transfer is now reduced to a single structural fact. Because the region transfer matrix is, by definition, the read-off of the virtual pullback, the matrix-structure hypothesis holds exactly when that pullback is of single-bond insertion form on f; the read-off then inverts the insertion and the round-trip closes the hypothesis. The image-preservation half is unconditional in the vertex-injective setting, so the pullback already realizes the transferred endpoint operator on every local tensor image of B at the in-region vertex. What remains is to show the pullback is of single-bond insertion form, which is the residual-independence the edge resonate identity supplies.[21]

The pin available from the closed-state transfer alone is non-circular but insufficient on its own. Evaluating the transferred endpoint operator of A on B's state vector at the endpoint physical leg recovers A's region-inserted coefficient; varying the region configuration at the endpoint vertex makes the leg range over the whole physical space while the state vector stays fixed, so the pin determines the endpoint operator's output on every state vector. But this only equates the operator with the read-off matrix *at*

[17] Formal declarations: `TNLean.PEPS.regionTransferMatrix`, `TNLean.PEPS.edgeTransferMatrix`, and `TNLean.PEPS.regionTransferMatrix_realizes_of_image`.

[18] Formal declaration: `TNLean.PEPS.physical_to_virtual_insertion`.

[19] Formal declarations: `TNLean.PEPS.span_stateOpenCoeff_eq_top`, `TNLean.PEPS.vertexComplementTensorInjective_of_isVertexInjective`, `TNLean.PEPS.range_localTensorMap_eq_span_stateVec`, `TNLean.PEPS.range_localTensorMap_eq_of_sameState`, and `TNLean.PEPS.regionInsertionOp_localProjectorAt_eq`.

[20] Formal declaration: `TNLean.PEPS.regionTransferRealizesAt_of_hform`.

[21] Formal declarations: `TNLean.PEPS.hform_of_isIncidentMatrixForm` and `TNLean.PEPS.localTensorMap_localVirtualOpOfPhysicalOpAt_regionInsertionOp`.

the single reference residual configuration used to define T_f; extending it to all residual configurations is the residual-independence the read-off must satisfy, and a single pin per region configuration cannot supply it. The closed-state transfer reads only one bond contraction (through R's endpoint vertex), so it cannot reconcile the two residual frames the way the edge proof does.[22]

The faithful route mirrors the edge proof's two-endpoint structure with R and its set complement $\mathrm{univ} \setminus R$ as the two contracted blocks of the single boundary edge f: the in-region endpoint vertex plays the role of one edge endpoint, the out-of-region endpoint vertex of f (the in-region endpoint of f for the complement region) plays the other, and the empty interior of the single edge f makes the middle block trivial. The region-inserted coefficient already carries this doubled boundary-configuration sum (one configuration read by R, one by the complement, coupled by the inserted matrix on f). The missing steps are: (i) a complement-side realization, the analogue of the realization of the region-inserted coefficient applied to the complement region with f as its boundary edge; (ii) a region resonate identity equating the v-side and complement-side realizations of A's region-inserted coefficient, transferred to B by SameState; (iii) inverting both endpoints through the blocked-region left inverses (using the blocked-region injectivity of R and of the complement); and (iv) a region reconcile forcing the two read-off matrices to coincide. This is the region-granularity port of the edge insertion-realization argument and remains the open obligation. Once it is supplied, the transfer datum and the per-edge gauge follow mechanically.

Step (i) now has a cast-free reading. The complement-side reading is the region-inserted coefficient read with the region and its set complement exchanged and the inserted matrix transposed on f. The exchange is the block-swap symmetry of the abstract two-block inserted coefficient, which stays over R's boundary-edge index set and so avoids the dependent-type reindexing $\mathrm{univ} \setminus (\mathrm{univ} \setminus R) = R$. What still requires that reindexing is factoring the complement block through the out-of-region endpoint vertex's local tensor map for the inversion of step (iii).[23]

Steps (ii) and the inversion infrastructure of step (iii) are now formalized. The out-of-region-endpoint pin reads the region-inserted coefficient through the out-of-region endpoint vertex by instancing the in-region pin at the set complement and casting back through the complement-side cast identity. Equating it with the in-region-endpoint pin gives the region resonate identity: the in-region and out-of-region endpoint operators of the first tensor, applied to the second tensor's closed state vectors, agree at the two endpoint legs. The region-inserted coefficient fac-

[22] Formal declarations: `TNLean.PEPS.regionInsertionOp_regionStateVec_pin` and `TNLean.PEPS.regionStateVec_update_vmem`.

[23] Formal declarations: `TNLean.PEPS.twoBlockInsertedCoeff_swap` and `TNLean.PEPS.regionInsertedCoeff_eq_complementTwoBlock`.

tors, as a function of the complement (resp. region) physical configuration, through the blocked-region tensor map of the set complement (resp. of the region); each blocked-region left inverse reads off the corresponding row function. What remains is the reconcile of step (iv): exhibiting the first tensor's region-inserted coefficient, as a function of the region physical configuration, in the image of the second tensor's region blocked tensor map (the membership that the resonate transfer across SameState supplies), then matching the two read-off row functions and concluding incident-matrix form of the virtual pullback through `TNLean.PEPS.hform_of_isIncidentMatrixForm`.

The remaining content of step (iv) is now isolated to a single predicate. Incident-matrix form of the virtual pullback, for every inserted matrix, assembles the realization bundle and hence the region insertion transfer datum and the per-edge gauge, all unconditionally on top of that predicate. The predicate names exactly the reconcile: the virtual pullback of the transferred in-region endpoint operator is of single-bond insertion form on the boundary edge. With it in both directions the per-edge gauge follows mechanically, so the open obligation is reduced to establishing this predicate from the region resonate identity and the two blocked-endpoint inversions.

The reconcile predicate is in turn reduced to a single coefficient identity. The leg-wise pin reads the in-region endpoint operator of the first tensor, applied to the second tensor's closed state vector, leg by leg as the first tensor's region-inserted coefficient scanned over the endpoint physical configuration. The closed state-vector coefficients span the full local virtual coefficient space at the in-region endpoint vertex, so a realization on the state vectors extends to all coefficients. Together these show: if for every inserted matrix on the first tensor's bond there is a matrix on the second tensor's bond whose region-inserted coefficient equals the first's at every physical configuration, then the reconcile predicate holds, with the read-off matrix the transpose of the matching matrix. The whole remaining obligation is therefore exactly this coefficient transfer: producing, for each inserted matrix, a matching matrix on the second tensor. This is the content the region resonate identity and the two blocked-endpoint inversions must supply; it is equivalent to the insertion-algebra isomorphism of the source lemma.[24]

The coefficient transfer is itself reduced to a single structural statement about a doubly-indexed transfer coefficient. The first tensor's region-inserted coefficient factors through both blocked endpoints of the second tensor at once: the v-side complement factorization gives the coefficient as a function of the complement physical configuration, and the symmet-

[24] Formal declarations: `TNLean.PEPS.regionInsertionOp_regionStateVec_pin_leg`, `TNLean.PEPS.regionInsertionOp_realizes_of_realizes_on_stateVec`, `TNLean.PEPS.regionInsertionOp_regionStateVec_eq_of_coeff_eq`, `TNLean.PEPS.isIncidentMatrixForm_of_coeff_eq`, and `TNLean.PEPS.regionResonateReconcile_of_coeff_transfer`.

ric region factorization (read off the complement-side cast identity and transported across the double complement) gives it as a function of the region physical configuration. Combining the two, the coefficient is the double sum of a transfer coefficient against the second tensor's region and complement blocked weights, with the transfer coefficient read off by the two blocked-endpoint left inverses; the required membership in the second tensor's complement block image holds because the region row is a finite linear combination of the v-side complement-image coefficients. Once the transfer coefficient is shown to have the incident-matrix coupling form of a single matrix on the boundary bond — coupling only the boundary-edge legs and contracting the residual legs by the identity — that matrix is the matching matrix the coefficient transfer asks for. This incident-matrix form of the transfer coefficient is the only remaining content; it is the region analogue of the source step $V = W$, forced by the resonate identity together with the full injectivity of both blocked endpoints.

The incident-matrix form of the transfer coefficient is now reduced, by a sorry-free chain, to the incident-matrix form of the virtual pullback itself. The transfer-coefficient column at a fixed complement boundary configuration is the second tensor's region blocked left inverse of the v-side row. The v-side row is the first tensor's in-region endpoint operator, from the inserted matrix, applied to the second tensor's region weight vector; since that weight vector is the second tensor's local tensor map of the region open coefficient, the unconditional image-preservation realization rewrites the v-side row as the second tensor's local tensor map of the virtual pullback applied to the region open coefficient. If the virtual pullback is of incident-matrix form, the second tensor's own in-region endpoint operator of that matrix realizes it, and the inner-sum realization expands the v-side row to the incident-matrix sum against the region blocked weights; the region blocked left inverse then collapses each weight to the standard basis configuration, reading off the incident-matrix coupling of the transfer coefficient. The reconcile predicate `TNLean.PEPS.RegionResonateReconcile` is exactly this incident-matrix form of the virtual pullback for every inserted matrix, so the whole open obligation is now this single predicate. Establishing it requires the out-of-region (complement-vertex) inversion of the region resonate identity, the region analogue of the right-endpoint inversion that supplies the residual-independence of the read-off matrix at the edge level; the in-region pin alone fixes the matrix only at the reference residual.[25][26][27][28][29]

The reconcile above is also reachable in a *block frame* that uses no single-vertex injectivity. The foundational input is the block-level image coincidence: under SameState, with the complement block blocked-tensor injective and positive bond dimensions, the range of the region blocked tensor map is the same for the two tensors, because both ranges equal the span of the partial states across the region cut, and the partial states depend only on the closed state coefficient. The same fact for the complement block follows by transporting blocked-tensor injectivity across the double complement. With these two image coincidences, the first tensor's region-inserted coefficient, read as a function of the region physical configuration, lies in the second tensor's region block image (region side), and the transferred row, as a function of the complement physical configuration, lies in the second tensor's complement block image (complement side); the two blocked left inverses read off the same transfer coefficient as in the vertex frame, and the first tensor's coefficient is the double sum of that transfer coefficient against the second tensor's two blocked weights. None of this uses single-vertex injectivity.

In the block frame the remaining content is isolated cleanly. By double blocked injectivity of the second tensor, the double-sum representation of a coefficient by a transfer coefficient is unique, so the transfer coefficient of an inserted matrix has the incident-matrix coupling form of a single bond matrix if and only if the first tensor's region-inserted coefficient of that inserted matrix equals the second tensor's of that bond matrix. The whole open obligation is therefore equivalent to the coefficient transfer alone — for every inserted matrix on the first tensor's bond, a matching bond

[25] Formal declarations (the sorry-free reduction): `TNLean.PEPS.transferCoeff_column_eq_regionBlockedLeftInverse_vSideRow`, `TNLean.PEPS.vSideRow_eq_localTensorMap_virtualPullback`, `TNLean.PEPS.vSideRow_eq_incidentSum_of_virtualPullback_incidentForm`, and `TNLean.PEPS.transferCoeff_eq_incidentForm_of_virtualPullback_incidentForm`.

[26] Formal declarations: `TNLean.PEPS.regionInsertedCoeff_eq_complement_blockedMap_B`, `TNLean.PEPS.regionInsertedCoeff_eq_region_blockedMap_B`, `TNLean.PEPS.regionBlockedTensorMap_doubleCompl`, `TNLean.PEPS.regionRowB_mem_range`, `TNLean.PEPS.regionInsertedCoeff_eq_doubleSum_transferCoeff`, `TNLean.PEPS.vSideRow_eq_region_blockedMap_transferCoeff`, and `TNLean.PEPS.regionInsertedCoeff_eq_of_transferCoeff_form`.

[27] Formal declarations: `TNLean.PEPS.RegionResonateReconcile`, `TNLean.PEPS.regionTransferRealizes_of_isIncidentMatrixForm`, `TNLean.PEPS.regionTransferRealizes_of_reconcile`, `TNLean.PEPS.regionInsertionTransfer_of_reconcile`, and `TNLean.PEPS.exists_regionEdgeGauge_of_reconcile`.

[28] Formal declarations: `TNLean.PEPS.regionInsertionOp_regionStateVec_pin_compl`, `TNLean.PEPS.region_resonate_identity`, `TNLean.PEPS.regionInsertedCoeff_eq_complement_blockedMap`, `TNLean.PEPS.regionInsertedCoeff_eq_region_blockedMap`, `TNLean.PEPS.regionBlockedLeftInverse_complement_regionInsertedCoeff`, and `TNLean.PEPS.regionBlockedLeftInverse_region_regionInsertedCoeff`.

[29] Formal declarations: `TNLean.PEPS.regionInsertedCoeff`, `TNLean.PEPS.regionInsertedCoeff_eq_regionRealizationSum`, `TNLean.PEPS.resonate_invert_right_endpoint`, `TNLean.PEPS.resonate_invert_left_endpoint`, `TNLean.PEPS.resonate_endpoint_coeff_reconcile`, `TNLean.PEPS.regionInsertionTransfer_of_realizes`, and `TNLean.PEPS.exists_regionEdgeGauge_of_transfer`.

matrix on the second tensor — with the matching matrix carried directly by the equivalence. The identity insertion is matched across SameState, so the transfer coefficient of the identity is already the incident-matrix coupling of the identity. The residual open content is the incident-matrix coupling form of the transfer coefficient for a general inserted matrix: the transfer coefficient must be local on the boundary bond, depending on the two boundary configurations only through their bond legs. This locality is the bond-level content of the source step $V = W$ and is the one fact the block-frame foundation does not supply, because the block image coincidence preserves only the full weight span and not the per-leg decomposition across the boundary bond.[30]

The block-frame foundation is now assembled around the edge-blocking triple. The union of two injective blocks is injective; applied to the blue and complement blocks of a one-edge blocking datum, this gives injectivity of the host block, the set complement of the red block, which every block-frame backbone lemma needs at the complement of the red region. With it, the four block injectivities of the two tensors at the red region and at its host are all read off a shared one-edge blocking datum, recording that the two tensors are blocked around the same edge into the same red, blue, and complement regions, together with positive bond dimensions. On top of these, the per-edge gauge is assembled from one predicate alone: bond locality of the transfer coefficient on the boundary edge, in both directions. Bond locality is proved equivalent to the coefficient transfer, so the open content of the block frame is now exactly this single predicate, with the per-edge gauge and all four injectivities derived from it and the shared blocking datum without single-vertex injectivity. The identity transfer of a single tensor is bond-local, the diagonal anchor of the cross-tensor transfer. The remaining open obligation is bond locality of the cross-tensor transfer for a general inserted matrix, the block-level content of the step $V = W$; it is the only fact between the assembled foundation and the per-edge gauge.[31]

The block-frame transfer kernel $K_M(\mu, \nu')$ is the double blocked left inverse of the first tensor's region-inserted coefficient of M through the sec-

[30] Formal declarations (block frame, sorry-free): `TNLean.PEPS.range_regionBlockedTensorMap_eq_of_sameState`, `TNLean.PEPS.regionBlockedTensorInjective_doubleCompl`, `TNLean.PEPS.range_regionBlockedTensorMap_compl_eq_of_sameState`, `TNLean.PEPS.regionInsertedCoeff_eq_blockTransferRow`, `TNLean.PEPS.blockTransferRow_mem_range`, `TNLean.PEPS.regionInsertedCoeff_eq_doubleSum_transferCoeff_block`, `TNLean.PEPS.doubleSum_kernel_injective`, `TNLean.PEPS.transferCoeff_eq_incidentKernel_iff_coeff_eq`, `TNLean.PEPS.transferCoeff_one_eq_incidentKernel_one`, `TNLean.PEPS.coeffTransfer_iff_transferCoeff_incidentForm`, and `TNLean.PEPS.exists_regionEdgeGauge_of_blockCoeffTransfer`.

[31] Formal declarations (block frame, sorry-free): `TNLean.PEPS.regionBlockedTensorInjective_union`, `TNLean.PEPS.regionBlockedTensorInjective_compl_red`, `TNLean.PEPS.SharedNormalEdgeBlockingData`, `TNLean.PEPS.IsBondLocalTransferKernel`, `TNLean.PEPS.bondLocal_iff_coeffTransfer`, `TNLean.PEPS.coeffTransfer_of_bondLocal`, `TNLean.PEPS.isBondLocalTransferKernel_self`, and `TNLean.PEPS.SharedNormalEdgeBlockingData.exists_regionEdgeGauge`.

ond tensor's region and complement blocks. Its bond locality `TNLean.PEPS.IsBondLocalTransferKernel` is the incident-matrix coupling form: the kernel couples the two boundary configurations only through their legs on the boundary edge f. The single remaining open step is the port of the edge resonate inversion to this block granularity, which hits one structural mismatch that is not a matter of length. The edge inversion `TNLean.PEPS.resonate_invert_right_endpoint` consumes the bond-contracted resonate identity `TNLean.PEPS.resonate_middle_inverted`, which pairs the left-endpoint and right-endpoint physical operators *indexed by the shared bond* k: the two endpoints are two single vertices sharing one bond, and the two operators act on the same physical space $\mathbb{C}^d$. In the block frame the two endpoints are the red region and its complement sharing the boundary edge f, and the only available factorizations of the first tensor's coefficient through the second tensor's complement block in the bond-f-paired form needed by the inversion — the v-side and σ-side blocked maps (`TNLean.PEPS.regionInsertedCoeff_eq_complement_blockedMap_B`, `TNLean.PEPS.regionInsertedCoeff_eq_region_blockedMap_B`) — carry the single-vertex hypothesis LinearIndependent (A_v) at the endpoint vertex of f. The block-image coincidence `TNLean.PEPS.range_regionBlockedTensorMap_eq_of_sameState` supplies the full-weight-span identity through the second tensor's whole region block as a single basis change, but it does not supply the per-f-leg bond pairing the inversion consumes; the only block-only complement-side reading of the coefficient is through the double left inverse K_M itself, which is the very kernel whose bond locality is the goal. The block-frame port therefore requires a vertex-injectivity-free bond-f-paired complement factorization of the coefficient before the edge inversion structure transfers, and constructing that factorization is the remaining open step.

The former vertex-injective bridge between these two reduction tracks has been retired. It did not supply bond locality in the normal setting, since its proof still assumed injectivity at the distinguished edge's endpoint vertex. Removing its two unused corollaries does not discharge this open step; the block-frame and coarse three-site arguments described here remain the active routes to it.

The coarse three-site route avoids the endpoint-vertex spanning entirely. The endpoint-vertex obstruction above is an artifact of running the comparison on the original single vertex at the end of the boundary edge. The source never uses that vertex. After blocking, the three sites of the chain are the blocks themselves, and the component family of each super-site is the blocked-region weight family, injective by the blocking hypothesis. Realizing this literally, the three blocks are assembled into a fresh tensor on a three-vertex complete graph: the red block at one vertex, the blue block at a second, the complement block at the third, with one super-bond for the distinguished edge, one for the red-to-complement crossings, and one for the blue-to-complement crossings, and a uniform padded physical dimension carrying every region's physical configuration through a fixed surjection. Each super-site's component family is the blocked-region weight family reindexed by the super-site's legs and read through the padded physical surjection; reindexing by a bijection and pulling back along a surjection both preserve linear independence, so each super-site is injective directly from the blocked-region injectivity of its block, with no single-vertex hypothesis. The single-vertex three-site comparison then applies to this coarse tensor verbatim: the two endpoint super-sites are injective by the red and blue blocking injectivities, the middle super-site is the single complement super-site, injective by the complement blocking injectivity, and the proven contraction-of-injective-tensors argument supplies the middle-block injectivity of the coarse chain. The coarse chain is therefore edge-blocked three-site injective at its distinguished super-bond with no single-vertex injectivity of the original tensor, and the proven injective inserted-coefficient correspondence runs on the coarse super-bond, producing, for every matrix inserted on the coarse super-bond of the first coarse tensor, a matrix on the second giving the same coarse inserted coefficient at every coarse physical configuration.[32]

Two obligations remain to descend the coarse super-bond correspondence to the original bond locality. The first is the state gluing: the coarse state coefficient must equal a positive constant times the original state coefficient of the assembled physical configuration, so that the original same-state hypothesis transports to the coarse tensors. The coarse state coefficient is the sum over the three super-bonds of the product of the three blocked-region weights, the three-block analogue of the two-block region gluing already formalized for one region against its complement. This gluing carries a compatibility requirement the present coarse frame does not yet record: each super-bond is incident to two super-sites, so its value is read by both incident per-site leg identifications, and those two identifications must send the shared super-bond value to the same original crossing-bond value. The distinguished super-bond is incident to the red and blue super-sites, the red-to-complement super-bond to the red and complement super-sites, the blue-to-complement super-bond to the blue and complement super-sites; for each, the two incident leg identifications must agree on the shared crossing bonds. Without this agreement the three blocked-region weights are not glued along their shared crossing edges and the coarse state coefficient does not factor through the original state coefficient. The second obligation is the descent identity: the coarse inserted coefficient on the distinguished super-bond must equal a positive constant times the original region-inserted coefficient of the same matrix,

[32] Formal declarations: `TNLean.PEPS.coarseGraph`, `TNLean.PEPS.CoarseBlockingFrame`, `TNLean.PEPS.CoarseBlockingFrame.coarseTensor`, `TNLean.PEPS.CoarseBlockingFrame.coarseTensor_isVertexInjective`, `TNLean.PEPS.CoarseBlockingFrame.coarseTensor_edgeBlockedThreeSiteInjective`, and `TNLean.PEPS.coarse_exists_edgeInsertedCoeff_eq`. The coarse construction and the injectivity transport are unconditional on top of the blocked-region injectivities, with no single-vertex spanning.

the three-block analogue of the landed identity reading the region-inserted coefficient of the identity as a constant times the state coefficient. With the super-bond identified with the original distinguished bond, this descent turns the coarse super-bond correspondence into the bond locality of the original transfer kernel and instantiates the per-edge gauge from the source hypotheses. Both obligations are state-gluing combinatorics over the crossing bonds, the three-block extension of the existing one-region-against-complement gluing; neither reintroduces single-vertex spanning.

The compatibility requirement of the first obligation is now recorded. The independent per-super-site leg identifications of the plain coarse frame are refined by a coherent coarse frame that fixes one bond model per coarse super-edge — an equivalence between the coarse bond and the original configurations on the edges crossing between the two incident regions — and requires each super-site's leg identification to factor through these shared bond models on its incident super-edges. The geometric content is that every boundary edge of a region crosses to exactly one partner region, so the two super-edges incident to a super-site carry exactly the two crossing bundles of its region's boundary. The payoff is the shared-super-bond agreement: two super-sites incident to a super-edge, fed matching super-bond values, read every shared crossing edge to the same original virtual leg. This is the well-posedness the state gluing needs — the three blocked-region weights are read at a single consistent assignment of the original crossing bonds.

The well-posedness layer is now fully reduced to the per-edge bond models. A partition hypothesis records that the three regions are pairwise disjoint and cover the vertex set, the geometry of the source edge blocking, detached from the tensor construction; from it the three regions form a three-block geometry, unlocking the landed three-block factorizations (the fused complement physical leg and the host vertex-product split over the complement of the red region). The partition gives the crossing classification: every boundary edge of a region crosses to exactly one partner region, so a red boundary edge is a red-to-blue or a red-to-complement crossing edge, and likewise at the blue and complement super-sites. Combined with the factoring fields, this expresses each super-site's leg identification of a coarse configuration entirely through the two bond models on its incident super-edges: the red boundary configuration reads each red boundary edge off the red-to-blue bond model (if it crosses to blue) or the red-to-complement bond model (if it crosses to the complement), and symmetrically at the blue and complement super-sites. For a single coarse configuration the three induced boundary configurations therefore agree on every shared crossing edge, the single-configuration form the merge collapse contracts against the original crossing edges.[33]

[33] Formal declarations (the well-posedness layer): `TNLean.PEPS.IsCrossingEdge`, `TNLean.PEPS.CrossingConfig`, `TNLean.PEPS.CoherentCoarseBlockingFrame`, `TNLean.PEPS.CoarseBlockingFrame.IsPartition`, `TNLean.PEPS.CoarseBlockingFrame.toThreeBlockGeometry`,

With this layer in place, the residual content of both obligations is the purely combinatorial three-region merge collapse: the triple sum over coarse virtual configurations of the boundary-coupled product of the three region weights refactors, through the partition of the original edges into the three regions' interiors and the three crossing bundles, as the product of the three regions' interior bond dimensions times the original closed-state coefficient of the assembled physical configuration. The single remaining mathematical step is the *global fiber-collapse bijection*: the bond models give a bijection between the coarse virtual configurations and the original inter-region crossing configurations, under which the triple sum's pairwise-agreement constraints become the boundary-agreement of an original configuration triple, and merging that triple into one global configuration (region-incident edges read from the region's own configuration, the rest free) collapses the sum to the closed-state coefficient with the three non-incident bond products as multiplicity. This is the three-region extension of the two-block merge/fiber collapse `TNLean.PEPS.stateCoeff_eq_regionComplement` of the landed one-region-against-complement gluing — a fiber count over the three regions' non-incident edges threading the per-edge bond models.

This collapse is now formalized. The bond models assemble into one product equivalence between coarse virtual configurations and triples of original inter-region crossing configurations, under which the coarse state sum reindexes as a sum over triples of original virtual configurations agreeing on the crossing edges; merging each agreeing triple into one global configuration (red-incident edges read the red configuration, the remaining blue-incident edges the blue, the rest the complement) collapses the sum to the closed-state coefficient, the fiber over a merged configuration being parameterised by the virtual indices each configuration is free to choose away from its own region. The constant is the product of the three regions' non-incident bond products. The same-state hypothesis transports through the collapse: the assembled physical configuration depends only on the shared regions and the merge-collapse constants coincide under matched bond dimensions. None of this reintroduces single-vertex spanning.[34]

What remained was the inserted-coefficient descent (the same collapse with a matrix carried on the coarse red-to-blue super-bond by its bond

`TNLean.PEPS.CoarseBlockingFrame.isCrossingEdge_red_blue_or_red_complement`, `TNLean.PEPS.CoarseBlockingFrame.isCrossingEdge_red_blue_or_blue_complement`, `TNLean.PEPS.CoherentCoarseBlockingFrame.legEquivRed_apply_eq`, `TNLean.PEPS.CoherentCoarseBlockingFrame.legEquivBlue_apply_eq`, `TNLean.PEPS.CoherentCoarseBlockingFrame.legEquivComplement_apply_eq`, and `TNLean.PEPS.CoherentCoarseBlockingFrame.legEquivRed_eq_legEquivBlue_on_rb_single`.

[34] Formal declarations: `TNLean.PEPS.crossingTripleEquiv`, `TNLean.PEPS.threeRegionSum_eq_agreeingTripleSum`, `TNLean.PEPS.triFiber_card`, `TNLean.PEPS.agreeingTripleSum_collapse`, `TNLean.PEPS.stateCoeff_coarseTensor_collapse`, and `TNLean.PEPS.coarseTensor_sameState_of_sameState`.

model) and the kernel closure (dividing the positive non-incident bond products and applying `TNLean.PEPS.bondLocal_iff_coeffTransfer` in both directions). Both are now landed; the per-edge gauge is delivered through the coherent-frame interface `TNLean.PEPS.exists_regionEdgeGauge_of_coherentFrames`, which discharges the three hypotheses `hbondAB`, `hbondBA`, and `hmul` of the abstract `TNLean.PEPS.SharedNormalEdgeBlockingData.exists_regionEdgeGauge` from the two coherent frames and the single-crossing hypothesis, returning the same gauge conclusion. The final assembly is recorded below.

The descent is not a single-edge assembly, because of a granularity gap recorded here. The coarse red-to-blue super-bond carries the *whole* bundle of red-to-blue crossing edges, not a single boundary edge: its bond model is an equivalence with the red-to-blue crossing configurations (`TNLean.PEPS.CrossingConfig`), and the coarse edge-inserted coefficient inserts a matrix on that whole bundle. The collapse with such a matrix therefore descends to a whole-bundle inserted coefficient (`TNLean.PEPS.redBundleInsertedCoeff`): a doubled red-boundary sum coupling the two configurations through their red-to-blue crossing labels (`TNLean.PEPS.redBoundaryRBCrossing`) while the red-to-complement crossings are contracted diagonally (`TNLean.PEPS.SameAwayFromRBBundle`). This coefficient is additive and homogeneous in the inserted matrix (`TNLean.PEPS.redBundleInsertedCoeff_add`, `TNLean.PEPS.redBundleInsertedCoeff_smul`). The relaxed crossing agreement of a triple under the inserted matrix — agreeing on the red-to-complement and blue-to-complement crossings, with the red-to-blue crossings left free for the matrix to couple — is recorded by `TNLean.PEPS.CrossTripleAgreesAwayRB`.

The open-bond reindexing of the descent is now landed. The coarse edge-inserted coefficient at the red-to-blue super-bond expands, through the open-bond form of the matrix insertion, as a sum over a coarse virtual configuration and a free right super-bond index of the matrix coupling their two super-bond values times the three region weights, the blue weight read through the override of the coarse configuration on the super-bond (`TNLean.PEPS.edgeInsertedCoeff_eq_pairSum`, `TNLean.PEPS.edgeInsertedCoeff_coarseTensor_eq_threeRegionSum`). Carrying the matrix through the coherent bond models reindexes this onto the relaxed crossing triples: the matrix becomes the bond-model-conjugated matrix (`TNLean.PEPS.bondModelMatrix`) at the two configurations' red-to-blue crossing labels, summed over triples agreeing away from the red-to-blue crossings (`TNLean.PEPS.mCoupledThreeRegionSum_eq_relaxedTripleSum`, built on the constraint-set collapse `TNLean.PEPS.pairConfig_constraint_set` and the per-pair triple expansion `TNLean.PEPS.perPair_threeRegionProduct_eq`). This is the matrix-carrying analogue of the closed-state reindexing `TNLean.PEPS.threeRegionSum_eq_agreeingTripleSum`.

The bundle-to-single-edge bridge is also landed. The single-edge predicate `TNLean.PEPS.IsBondLocalTransferKernel` that the per-edge gauge consumes, and the equivalence `TNLean.PEPS.bondLocal_iff_coeffTransfer` that reduces it to a single-edge coefficient transfer, are stated for one boundary edge f. When the red region crosses to the blue region across the single distinguished edge e — the source's geometry, where the red sites surround one endpoint and the blue sites the other — the whole red-to-blue crossing bundle is one virtual leg on e (`TNLean.PEPS.singletonCrossingEquiv`), the bundle agreement is the single-edge agreement `TNLean.PEPS.SameAwayFromBond` on e (`TNLean.PEPS.sameAwayFromRBBundle_iff_sameAwayFromBond`), and the whole-bundle red inserted coefficient of the carried bond matrix is the single-edge `TNLean.PEPS.regionInsertedCoeff` of that matrix on e (`TNLean.PEPS.redBundleInsertedCoeff_singleton`). The single-crossing hypothesis is the explicit assumption that the red-to-blue crossings are exactly the singleton $\{e\}$, which the source's coordinate blocking discharges (`TNLean/PEPS/NormalBlocking.lean:251--264,367--379`).

With these two landed pieces, the descent on both tensors (sharing the non-incident bond products under matched bond dimensions, by `TNLean.PEPS.coarseTensor_sameState_of_sameState`), the coarse coefficient transfer `TNLean.PEPS.coarse_exists_edgeInsertedCoeff_eq`, the surjectivity `TNLean.PEPS.coarseProj_surjective`, and the positivity of the non-incident bond products would give the single-edge coefficient transfer, hence `TNLean.PEPS.IsBondLocalTransferKernel` in both directions and the per-edge gauge.

The *relaxed-triple merge collapse* is now landed (`TNLean.PEPS.relaxedTripleSum_collapse`), and with it the whole inserted-coefficient descent (`TNLean.PEPS.edgeInsertedCoeff_coarseTensor_descent`): the coarse edge-inserted coefficient at the red-to-blue super-bond equals a constant times the whole-bundle red inserted coefficient `TNLean.PEPS.redBundleInsertedCoeff` of the bond-model-conjugated matrix, with the host physical leg the fused blue/complement leg. The collapse is the matrix-carrying analogue of `TNLean.PEPS.agreeingTripleSum_collapse`, a *two-sided* merge: the red configuration ζ_r merges into the red-boundary index μ (collapsing to `TNLean.PEPS.regionBlockedWeight` of μ with no fiber multiplicity, the blocked-region weight already summing the red product over every configuration carrying the red boundary label), while the blue and complement configurations (ζ_b, ζ_c), agreeing on the blue-to-complement crossings, merge into the host configuration `TNLean.PEPS.hostMerge` carrying the host boundary index ν over univ $\setminus$ red. The bond-model-conjugated matrix couples μ and ν through their red-to-blue crossing labels (`TNLean.PEPS.redBoundaryRBCrossing`), and the red-to-complement crossing agreement is the bundle agreement `TNLean.PEPS.SameAwayFromRBBundle`.

The merged summand $\mathrm{matrix}(\mu_{\mathrm{rb}}, \nu_{\mathrm{rb}}) \cdot \mathrm{redProd}\, \zeta_r \cdot \mathrm{hostProd}(\mathrm{hostMerge})$ is `TNLean.PEPS.relaxedTriple_summand_eq`: the complement and blue vertex products read the host merge unchanged (`TNLean.PEPS.complPro `d_hostMerge`, `TNLean.PEPS.blueProd_hostMerge`), so the host vertex product of the merge, read with the fused blue/complement physical leg, is the complement product against the blue product (`TNLean.PEPS.hostProd_hostMerge_eq`); the two boundary labels and the matrix coupling are identified by `TNLean.PEPS.sameAwayFromRBBundle_hostMerge`, `TNLean.PEPS.redBoundaryRBCrossing_hostMerge`, and `TNLean.PEPS.crossingLabel_eq_redBoundaryRBCrossing`. The host side collapses through the host-merge fiber count `TNLean.PEPS.hostMergeFiber_card`: the relaxed pairs agreeing on the blue-to-complement crossings whose merge is a fixed configuration biject with the free virtual indices, the fiber multiplicity being `TNLean.PEPS.hostMergeFiberProd` (the complement non-incident bond product times the free blue bond product), a function of the bond dimensions alone and hence shared across the two tensors under matched bond dimensions. The fiberwise collapse `TNLean.PEPS.hostMerge_fiberwise_collapse` and the relaxed agreement split `TNLean.PEPS.crossTripleAgreesAwayRB_iff` wire the two-stage grouping against the configuration expansion `TNLean.PEPS.redBundleInsertedCoeff_bondModelMatrix_eq_configSum` of the target coefficient. The descent uses no single-vertex spanning.

The final assembly is now landed. Under the single-crossing hypothesis the whole inserted-coefficient descent reads, through the bond-model and the single-leg crossing equivalence, as a constant times the single-edge region-inserted coefficient on e (`TNLean.PEPS.edgeInsertedCoeff_coarseTensor_descent_single`), the constant the shared positive host-merge fiber product (`TNLean.PEPS.hostMergeFiberProd_pos`, `TNLean.PEPS.hostMergeFiberProd_eq`). Cancelling that constant after the descent on both coherent frames, and feeding the coarse coefficient transfer and the surjectivity of the descent's physical legs (`TNLean.PEPS.exists_descent_legs`), gives the single-edge coefficient transfer in both directions (`TNLean.PEPS.exists_regionInsertedCoeff_eq_sharedRegion`, `TNLean.PEPS.exists_regionInsertedCoeff_eq_sharedRegion_symm`), hence the two bond-local transfer kernels (`TNLean.PEPS.isBondLocalTransferKernel_of_coherentFrames`). The forward multiplicativity is the concrete single-edge transfer's multiplicativity, the coarse edge transfer matrix conjugated by the two bond models (`TNLean.PEPS.regionEdgeTransfer`, `TNLean.PEPS.regionEdgeTransfer_mul`), carried through `TNLean.PEPS.regionInsertedCoeff_injective` (`TNLean.PEPS.coeffTransferMap_eq_regionEdgeTransfer`). With the two kernels and the forward multiplicativity, the per-edge gauge follows (`TNLean.PEPS.exists_regionEdgeGauge_of_coherentFrames`): the forward per-edge matrix transfer on e is conjugation by an invertible gauge matrix and the bond dimensions on e coincide. The whole assembly uses no single-vertex injectivity of the original tensor; the only injectivity input is the blocked-region injectivity of the three regions, carried by the coherent coarse blocking frame. This closes, modulo review, the block-level step

$V = W$ and the forward multiplicativity — the last residual content of this obligation — for two coherent frames sharing the three regions, the bond dimensions, and the single distinguished crossing edge.

The coherent-frame interface is now fed from a one-edge blocking datum. The coherent coarse blocking frame of a one-edge blocking datum over the concrete region-injectivity predicate of a tensor is assembled (`TNLean.PEPS.coherentFrameOfBlockingData`), and two such data over the two tensors, sharing the three regions and the single distinguished crossing edge, deliver the per-edge gauge: the bond dimensions on e coincide and the forward per-edge matrix transfer on e is conjugation by an invertible matrix (`TNLean.PEPS.exists_regionEdgeGauge_of_blockingData`), with no single-vertex injectivity.

The single-crossing hypothesis is discharged from the coordinate geometry of the square lattice. For the translated horizontal edge blocking the red block occupies the columns $x_0, x_0{+}1$ and the rows $y_0{+}2, \ldots, y_0{+}4$ and the blue block the columns $x_0{+}2, \ldots, x_0{+}4$ and the rows $y_0{+}1, y_0{+}2$, two adjacent rectangles meeting at the single nearest-neighbour pair $(x_0{+}1, y_0{+}2)$–$(x_0{+}2, y_0{+}2)$, the distinguished edge; the rotated picture holds for the vertical blocking. Hence an edge crosses between the red and blue blocks if and only if it is the distinguished edge (`TNLean.PEPS.isCrossingEdge_normalSquareHorizontalTranslatedEdge`, `TNLean.PEPS.isCrossingEdge_normalSquareVerticalTranslatedEdge`), with interior-data wrappers tying the single-crossing hypothesis to the cover-free interior blocking data. The region projections of the interior data are exposed syntactically as the coordinate blocks (`TNLean.PEPS.normalSquareHorizontalTranslatedEdge_blockingDatum_interior_red` and companions).

What remains for the per-edge gauge family is a decidable-equality coherence bridge. The square-lattice geometry layer builds its blocking data with the canonical product decidable equality on $\mathrm{Fin}\, w \times \mathrm{Fin}\, h$, while the gauge interface lives over the lattice's linear order and uses the order's decidable equality. The two instances are equal as a subsingleton, so they carry the same mathematics, but they are not definitionally equal, and the two layers meet exactly when an interior datum is fed to the gauge interface. Feeding the gauge per interior edge therefore requires aligning the two decidable-equality instances — by transporting the datum across the subsingleton equality, or by a uniform instance choice across the geometry and gauge files. This is a finite engineering step with no further mathematical content: the gauge of a one-edge datum (`TNLean.PEPS.exists_regionEdgeGauge_of_blockingData`) and the single-crossing discharge are already in place. The orientation-uniform reduction (obligation 6) and the boundary geometry (obligation 4) remain the open mathematical pieces above this bridge.

5. In the translationally invariant case, show that the edge gauges obtained

after blocking reduce to one horizontal gauge X and one vertical gauge Y, up to scalar.

The carrier of this reduction is now formalized. A square-lattice bond-dimension function is orientation uniform when every horizontal edge has one dimension and every vertical edge has another; given that, a single horizontal matrix and a single vertical matrix determine a gauge family by transport across the bond-dimension equalities, and a gauge family of this shape is called orientation uniform. The absorption step incorporates such an orientation-uniform family into the second tensor, preserving the orientation-uniform bond dimensions and vertex injectivity; the post-absorption edge-insertion equality is taken as the conditional kernel input, in the shape produced unconditionally in the vertex-injective case.[35]

The uniqueness-up-to-scalar step is now formalized. The conjugating matrix realizing a forward per-edge transfer is determined up to a multiplicative constant: if two invertible matrices induce the same conjugation on every bond matrix then their quotient commutes with the whole bond matrix algebra, hence lies in its center, the scalars, so the two differ by a nonzero constant. Reading this through the bond-dimension reindexing gives the same statement for the per-edge transfer map of an edge: two gauges with the same transfer map differ by a constant. This is the source's statement that the horizontal and vertical matrices are unique up to a multiplicative constant, at the level of a single edge.[36]

Because each per-edge gauge is fixed only up to a constant, the reduction to one horizontal and one vertical matrix holds only up to a per-edge scalar: a gauge family is orientation uniform up to scalars when each edge gauge is the transported reference matrix rescaled by a nonzero per-edge constant. The assembly across the two orientation classes is formalized: collecting, on each class, the per-edge "agree with the reference up to a constant" facts produces one orientation-uniform-up-to-scalar family, with the exact orientation-uniform family as the all-constants-one specialization.[37]

The per-edge constants are absorbed at the absorption step when they are vertex balanced. A per-edge scalar rescaling whose oriented product around every vertex is one is a balanced edge-scalar reweighting, and such a reweighting leaves the absorbed tensor unchanged. Hence the scalar-carrying orientation-uniform gauge and the underlying exact orientation-

[35] Formal declarations: `TNLean.PEPS.SquareLatticeUniformBondDim`, `TNLean.PEPS.orientationUniformGauge`, `TNLean.PEPS.IsOrientationUniformGaugeFamily`, `TNLean.PEPS.exists_orientationUniformGaugeFamily`, `TNLean.PEPS.tiAbsorb`, `TNLean.PEPS.isVertexInjective_tiAbsorb`, `TNLean.PEPS.tiNormalGaugeAbsorption`, and `TNLean.PEPS.exists_postAbsorption_of_vertexInjective`.

[36] Formal declarations: `TNLean.PEPS.gl_conj_unique_scalar` and `TNLean.PEPS.edgeGauge_unique_scalar`.

[37] Formal declarations: `TNLean.PEPS.IsOrientationUniformGaugeFamilyModScalar`, `TNLean.PEPS.isOrientationUniformGaugeFamilyModScalar_of_isOrientationUniform`, and `TNLean.PEPS.isOrientationUniformGaugeFamilyModScalar_of_classAgreement`.

uniform gauge incorporate into the same modified second tensor, so the conditional square-lattice theorem may consume an exact orientation-uniform gauge even though the blocking delivers one only up to balanced constants.[38]

The periodic lattice on which this transport lives is now formalized, together with the covariance of the blocking machinery under it. The discrete torus carries cyclic nearest-neighbour edges on which the coordinate translations are graph automorphisms, a tensor is translation invariant when fixed by transport along every translation, and such a tensor has orientation-uniform bond dimensions on each class.[39] The blocking data themselves transport: the blocked-region weight of a tensor transported along a graph isomorphism is the original weight reindexed by the edge action and the vertex bijection, so blocked-region injectivity transports, and a one-edge blocking datum transports to a datum for the transported tensor at the image edge with the single red-to-blue crossing carried across. Specialized to a translation-invariant tensor, this carries one reference blocking datum at the reference edge of an orientation class to a datum at every translate of that edge, for the same tensor.[40] The reference blocking datum at each class's reference edge is now constructed concretely. For a vertex-injective tensor with positive bond dimensions every finite vertex region is injective, so the singleton endpoint blocks and their complement form a one-edge blocking datum with no rectangular geometry: the left endpoint is the red block, the right endpoint the blue block, the rest of the torus the complement, and the only red-to-blue crossing is the edge itself. This supplies the reference datum at the reference horizontal and vertical edges directly, with no wraparound-avoiding rectangle bounds, since singleton blocks need no room to be injective once the tensor is vertex injective.[41] **Scope restriction (NormalTorusRectangleInjectivity-Hypotheses):** this singleton datum is a vertex-injective specialization of the normality hypothesis in Theorem 3 of arXiv:1804.04964, lines 1449–1452. The source-faithful torus reference datum uses the rectangular injectivity hypotheses directly: the reference red and blue blocks are the two source rectangles adjacent to the distinguished edge, and the complement is the union of the surrounding bands and fillers.[42] The per-edge gauge

[38] Formal declarations: `TNLean.PEPS.gaugeEquivModEdgeScalars_of_matrixScalar` and `TNLean.PEPS.absorbEdgeGauges_eq_of_matrixScalar`.

[39] Formal declarations: `TNLean.PEPS.torusGraph`, `TNLean.PEPS.translate`, `TNLean.PEPS.IsTorusTranslationInvariant`, `TNLean.PEPS.TorusUniformBondDim`, and `TNLean.PEPS.torusUniformBondDim_of_translationInvariant`.

[40] Formal declarations: `TNLean.PEPS.regionBlockedWeight_transport`, `TNLean.PEPS.regionBlockedTensorInjective_transport`, `TNLean.PEPS.transportBlockingDataAlong`, `TNLean.PEPS.isCrossingEdge_transportBlockingDataAlong`, `TNLean.PEPS.translateBlockingData`, and `TNLean.PEPS.isCrossingEdge_translateBlockingData`.

[41] Formal declarations: `TNLean.PEPS.singletonEdgeBlockingData`, `TNLean.PEPS.isCrossingEdge_singletonEdgeBlockingData`, `TNLean.PEPS.torusHorizontalReferenceBlockingDatum`, and `TNLean.PEPS.torusVerticalReferenceBlockingDatum`.

[42] Formal declarations: `TNLean.PEPS.torusHorizontalRectangleBlockingDatum`, `TNLean.P`

interface applies verbatim on the torus (it is stated over a general finite ordered vertex set), and the orientation-uniform reduction is ported to the torus, so the assembly above a class-agreement input is in place there too.[43]

The covariance of the region-inserted coefficient is now established, and with it the determinacy that closes the class-agreement input above the per-edge gauge of obligation 5. The region-inserted coefficient of a tensor transported along a graph isomorphism, over the image region with the inserted matrix carried to the image bond and the boundary edge to its image, equals the original coefficient over the original region; the boundary configurations reindex by the edge action, the matrix coupling and the away-from-the-bond agreement are preserved, and the region and complement weights both transport. No endpoint orientation enters, since the coefficient inserts the matrix between the two single boundary values on the edge. [44] Specialized to a translation-invariant pair, this carries the coefficient identity at a reference edge to the same identity at every translate of that edge, for the same two tensors, since the tensors are fixed by transport. [45]

The non-constructive transfer map of the gauge interface is determined by the coefficient identity it satisfies: with the second tensor's region and complement injective and positive bond dimensions, two matrix maps realizing the same coefficient identity coincide. Hence the per-edge transfer map at a translate equals the reference transfer map carried across the translation, two independently chosen per-edge gauges realize the same conjugation map, and the per-edge uniqueness up to scalar supplies the constants; gathering the two orientation classes yields the orientation-uniform-up-to-scalar family. [46] The algebraic reduction from matched coefficient identities to the conjugation covariance is now in place. The per-edge gauge at a torus edge carries its defining coefficient identity, so the gauge conjugation itself satisfies the identity. Reindexing across a bond-dimension equality commutes with conjugation, identifying the transported reference matrix

`EPS.torusVerticalRectangleBlockingDatum`, `TNLean.PEPS.isCrossingEdge_torusHorizontalRectangleBlockingDatum`, and `TNLean.PEPS.isCrossingEdge_torusVerticalRectangleBlockingDatum`.

[43]Formal declarations: `TNLean.PEPS.exists_regionEdgeGauge_torus`, `TNLean.PEPS.torusOrientationUniformGauge`, `TNLean.PEPS.IsTorusOrientationUniformGaugeFamilyModScalar`, `TNLean.PEPS.isTorusOrientationUniformGaugeFamilyModScalar_of_classAgreement`, and `TNLean.PEPS.isTorusOrientationUniformGaugeFamilyModScalar_of_translationInvariant`.

[44]Formal declarations: `TNLean.PEPS.regionComplementWeight_transport`, `TNLean.PEPS.sameAwayFromBond_regionBoundaryConfigMap`, and `TNLean.PEPS.regionInsertedCoeff_transport`.

[45]Formal declarations: `TNLean.PEPS.bondDim_boundaryEdgeMap_translate`, `TNLean.PEPS.regionInsertedCoeff_congrTensor`, and `TNLean.PEPS.regionInsertedCoeff_translate_coeffIdentity`.

[46]Formal declarations: `TNLean.PEPS.regionInsertedCoeff_transferMap_unique`, `TNLean.PEPS.regionTransferMap_eq_of_coeffIdentities`, and `TNLean.PEPS.isTorusOrientationUniformGaugeFamilyModScalar_of_conjCovariance`.

with the conjugation by the reindexed reference. Two gauges realizing the same coefficient identity by conjugation induce the same conjugation map on every bond matrix, by the transfer-map determinacy. These supply, edge by edge, the conjugation coincidence the orientation-uniform selection consumes.[47] What remains is the geometric production of the two matched coefficient identities at every edge: reading the per-edge gauge off the reference blocking datum transported along each class translation, and matching its region and boundary edge with those of the transported reference identity. The transported blocking datum reads its three blocks from the translation, which may reorder the edge endpoints and so place the reference red region in the blue slot; the orientation-class matching must account for this branch. This construction still depends on the open per-edge gauge of obligation 5 for the transfer maps themselves.

The endpoint-reordering branch is now handled at the level of the gauge-interface inputs. The transported datum's complement block is always the image of the reference complement, and its red block is one of the two image blocks; whichever image block plays the red role, that block is injective and the union of the remaining two covers the host region of the red block and is injective under union closure. So both injectivity inputs the per-edge gauge consumes are available on both branches, with no case split, and the region-inserted coefficient is endpoint-symmetric. What is left is to route the orientation-class matching through the image of the reference red region rather than the possibly-swapped red slot, which the region-polymorphic coefficient covariance already supplies.[48]

The reference blocking datum is now produced from the source's own rectangular-injectivity hypotheses on the torus, with no single-vertex injectivity. Contiguous torus rectangles read the half-open coordinate intervals through the value embedding, so wraparound-avoiding offsets recover the contiguous block and full-coordinate bands cover the whole range; the 2×3 and 3×2 shape predicates, the native rectangular-injectivity hypotheses, and the sliding-window tiling are the torus ports of the open-lattice development. The horizontal and vertical reference edges carry the red (removed source rectangle), blue (the other removed source rectangle), and complementary (the band-and-filler cover) blocks; all three are injective by rectangular injectivity and union closure, they partition the torus, and the cyclic-step value lemmas read a wraparound-free step as the ordinary step, so the single red-to-blue crossing is the reference edge. This is the source-faithful replacement for the vertex-injective singleton scaffolding.[49]

[47] Formal declarations: `TNLean.PEPS.exists_regionEdgeGauge_torus_coeff`, `TNLean.PEPS.regionInsertedCoeff_translate_coeffIdentity_conj`, `TNLean.PEPS.gaugeConj_eq_of_coeffIdentities`, and `TNLean.PEPS.gaugeConj_eq_of_coeffIdentities_target`.

[48] Formal declarations: `TNLean.PEPS.transportBlockingDataAlong_complement`, `TNLean.PEPS.regionBlockedTensorInjective_transportBlockingDataAlong_red`, and `TNLean.PEPS.regionBlockedTensorInjective_host_transportBlockingDataAlong`.

[49] Formal declarations: `TNLean.PEPS.NormalTorusRectangleInjectivityHypotheses`,

Feeding this faithful datum to the per-edge gauge engine yields, at the reference edge, the per-edge conjugation gauge realizing the region-insertion coefficient identity, with the second tensor's red and host injectivities supplied by its own rectangle hypotheses and union closure. The conjugation-covariance input the orientation-uniform selection consumes is packaged as a per-edge coefficient-identity witness — a region with single boundary edge e, the second tensor's region and host injectivity, and the two coefficient identities the per-edge gauge and the transported reference gauge realize. Each witness yields the conjugation coincidence by the transfer-map determinacy, and gathering the two orientation classes assembles the orientation-uniform-up-to-scalar family. The reference edge inhabits this witness interface directly.

The translation production is now carried out. The reference coefficient identity, realized by conjugation with the reference gauge, transports along each class translation to the same identity at every edge of the class, realized by conjugation with the reference gauge carried across the bond-dimension reindexing. The translated region is the image of the reference region; the second tensor's image region and its host are injective there by transport of the reference injectivities, since the second tensor is fixed by the translation; and an arbitrary region or complement configuration on the image region is the transport of a unique configuration on the reference region. Composing the two reindexings of the transported gauge collapses to the single transported orientation matrix the selection expects, so the per-edge witness against that matrix holds at every edge. Gathering the horizontal and vertical witness families through the orientation-uniform selection produces, from the source's rectangle hypotheses on a translation-invariant pair, the orientation-uniform-up-to-scalar family unconditionally: the per-edge gauge family is the single reference matrix per class transported to every edge, the source's "X and Y, the same matrix on all horizontal (vertical) edges", with the per-edge scalars all one.[50]

One strengthening remains. The family produced above takes the per-edge gauge to be the transported reference matrix itself, so the witness's two coefficient identities are the same single identity and the per-edge scalars are all one. The source's stronger claim is that *any* per-edge gauge independently produced by the edge-blocking engine at each edge agrees with the transported reference up to a scalar. Reading that independent engine gauge over the image of the reference region requires a blocking da-

`TNLean.PEPS.NormalTorusRectangleInjectivityHypotheses.horizontalEdgeComplement_injective`, `TNLean.PEPS.torusHorizontalRectangleBlockingDatum`, `TNLean.PEPS.torusVerticalRectangleBlockingDatum`, `TNLean.PEPS.isCrossingEdge_torusHorizontalEdge`, and `TNLean.PEPS.isCrossingEdge_torusVerticalEdge`.

[50]Formal declarations: `TNLean.PEPS.regionInsertedCoeff_translate_coeffIdentity_conj`, `TNLean.PEPS.edgeCoeffIdentityWitness_translate`, `TNLean.PEPS.edgeCoeffIdentityWitness_translateUniform`, `TNLean.PEPS.exists_edgeCoeffIdentityWitness_horizontalFamily`, `TNLean.PEPS.exists_edgeCoeffIdentityWitness_verticalFamily`, and `TNLean.PEPS.isTorusOrientationUniformGaugeFamilyModScalar_torus_rectangle`.

tum at the translated edge whose red block is the image of the reference red block; but the lexicographic edge convention reorders the endpoints under a wraparound translation, so the transported datum's red block is the image of the reference *blue* block on that branch, and the reference coefficient identity — tied to the reference red region — does not transport to it. Closing the independent-gauge strengthening therefore needs a branch-free reference identity (one stated over the reference region's host orientation as well), which the present reference engine does not supply. The unconditional family above does not depend on this strengthening.

The final region comparison setup is also formalized: a region and its set complement are wrapped as the two blocks consumed by the two-injective comparison, and from matched region-inserted coefficients of the two tensors the region blocks are scalar proportional, the region analogue of the one-vertex-versus-complement comparison. For vertex-injective tensors the four block-injectivity hypotheses are discharged uniformly, leaving the region-inserted-coefficient equality as the only conditional input.[51]

The region-inserted-coefficient equality consumed by the region comparison is the *absorbed* form: A inserting M on f matches $\widetilde{B} = \mathrm{applyGauge}\,B\,X$ inserting the same M on f. The per-edge gauge delivered by the blocking is, however, the *conjugation* form: A inserting M matches B inserting $X^{\mathsf{T}}M(X^{-1})^{\mathsf{T}}$ on f. The bridge between the two is the region-granularity port of the open-edge gauge cancellation `TNLean.PEPS.edgeInsertedCoeff_applyGauge`: inserting M on the gauged tensor cancels every interior gauge of the region and of its complement and conjugates M by the gauge on the single boundary edge f. This bridge is now formalized as `TNLean.PEPS.regionInsertedCoeff_applyGauge`.

The proof factors through a region-to-edge identity. First, the double-global-configuration form of the region-inserted coefficient (`TNLean.PEPS.regionInsertedCoeff_eq_doubleSum`) collects its region and complement vertex weights into one global vertex product over all vertices, the region side reading the first configuration of the agreeing pair and the complement side the second (`TNLean.PEPS.pairOuter`), with the physical legs assembled; this holds both for the gauged tensor (`TNLean.PEPS.regionComplProd_gauge_eq`) and gauge-free (`TNLean.PEPS.regionComplProd_eq`). The outer reading `TNLean.PEPS.pairOuter` of an agreeing pair is consistent off f (`TNLean.PEPS.pairOuter_isConsistentOff`): only the single boundary edge f keeps two distinct endpoint indices. The agreeing pairs over a fixed consistent-off-f configuration are parameterised by the free non-boundary legs, so each fiber has the interior bond multiplicity (`TNLean.PEPS.pairOuterFiber_card`), and summing a function of `TNLean.PEPS.pairOuter` over the agreeing pairs is that multiplicity times

[51] Formal declarations: `TNLean.PEPS.regionComplement_comparison` and `TNLean.PEPS.regionComplement_comparison_of_vertexInjective`.

the sum over consistent-off-f configurations (`TNLean.PEPS.sum_pairOuter_fiber_collapse`). Identifying the latter with the edge-inserted coefficient at f of the assembled physical configuration gives the region-to-edge identity `TNLean.PEPS.regionInsertedCoeff_eq_smul_edgeInsertedCoeff`: the region/complement contraction is the single-bond cut at f up to the gauge-invariant interior bond multiplicity, with the inserted matrix oriented to the edge convention by `TNLean.PEPS.regionEdgeOrient` (identity when f's left endpoint lies in R, transpose otherwise). The edge gauge cancellation `TNLean.PEPS.edgeInsertedCoeff_applyGauge` then transports across the identity, giving the bridge with the conjugation by X_f^{T} on the orientation-corrected matrix.

The conjugation-to-absorbed conversion is now formalized, both at a single comparison region and edge-level (so region-independent). At a region R with boundary edge f, the engine's conjugation-form coefficient identity A inserting M equals B inserting $Z\,M\,Z^{-1}$ becomes the absorbed equality A inserting M equals applyGauge $B\,X$ inserting the same M, with the absorbing matrix at f the orientation-adapted transpose of the engine gauge (`TNLean.PEPS.absorbedBoundaryGauge`: Z^{T} when f's left endpoint lies in R, Z^{-1} otherwise, so the bridge's $X^{\mathsf{T}} \cdot \text{orient} \cdot (X^{-1})^{\mathsf{T}}$ conjugation reproduces the engine's $Z \cdot\, \cdot Z^{-1}$ conjugation). The matrix reconciliation is `TNLean.PEPS.regionEdgeOrient_absorbedBoundaryGauge`; the per-region conversion is `TNLean.PEPS.regionInsertedCoeff_eq_applyGauge_of_conjIdentity`. Cancelling the shared positive interior multiplicity in the region-to-edge identity reduces the region absorbed equality to the edge-level absorbed equality at f over the bare edge (`TNLean.PEPS.edgeInsertedCoeff_eq_applyGauge_of_region`), and the converse multiplies it back up to the region absorbed equality at *every* region with f on its boundary (`TNLean.PEPS.regionInsertedCoeff_eq_applyGauge_of_edge`). The two directions are the route note's region-independence: the absorbed plain equality over any comparison region follows from the single edge-level identity at the boundary edge, which is what the region comparison `TNLean.PEPS.regionComplement_comparison` consumes.

On the torus the per-edge coefficient identity is produced at a single boundary edge of each *red blocking region* of the orientation-uniform family (`TNLean.PEPS.exists_regionEdgeGauge_torus_coeff`, `TNLean.PEPS.isTorusOrientationUniformGaugeFamilyModScalar_torus_rectangle`). The per-vertex relation $A_v = \lambda \cdot \text{gaugeVertex}\,B\,X\,v$ that the torus Fundamental Theorem (`TNLean.PEPS.fundamentalTheorem_normalTorusPEPS`) takes as the hypothesis `hPV` still needs two pieces above the region-independence just recorded. First, the absorbing matrix `TNLean.PEPS.absorbedBoundaryGauge` is the orientation-adapted transpose of each *per-edge* engine gauge Z_e, so the absorbed equality holds against a per-edge absorbing family rather than against the single orientation-uniform $X = \text{torusOrientationUniformGauge}$ that `TNLean.PEPS.isTorusOrientationUniformGaugeFamilyModScalar_torus_rectangle` returns and that `hPV` is stated against; reconciling the per-edge absorbing family with that orientation-uniform X (the transpose/orientation bookkeeping across the two orientation classes) is the first open piece. Second, feeding the absorbed equality to `TNLean.PEPS.regionComplement_comparison` at the source comparison regions R and S (which differ in one site, both blocked-tensor injective with injective complements) gives $A_R \propto \widetilde{B}_R$ and $A_S \propto \widetilde{B}_S$; quotienting the two region proportionalities to isolate the single differing vertex is the second open piece. The block-granularity one-site quotient that performs this isolation without single-vertex injectivity — the region analogue of `TNLean.PEPS.perVertexScalar`, whose present form rests on `TNLean.PEPS.one_vertex_complement_comparison` and edge-level vertex injectivity — needs a factorization of the blocked-region weight of withSite v through the blocked-region weight of withoutSite v and the single vertex tensor at v (the one-vertex analogue of `TNLean.PEPS.regionBlockedWeight_eq_localTensorMap`).

The foundational layer of this quotient is now landed. Writing the differing regions as $S = \mathrm{insert}\, v\, R$ with $v \notin R$, the vertex product over the sites of S factors as the inserted-site tensor times the vertex product over R (`TNLean.PEPS.prod_region_insert_split`), and this lifts to the constrained blocked-weight sum (`TNLean.PEPS.regionBlockedWeight_insert_eq_sum_split`): the blocked weight of S is the μ-constrained sum of A.component v against the residual vertex product over R. The boundary-configuration half is also landed: an R-boundary edge not incident to v is an S-boundary edge (`TNLean.PEPS.isRegionBoundaryEdge_insert_of_not_incident`), one incident to v becomes internal to S, and the boundary label of R is folded from the S-boundary label away from v and the local virtual configuration at v on the v-incident edges (`TNLean.PEPS.boundaryLabelOfInsert`, `TNLean.PEPS.regionBoundaryLabel_eq_boundaryLabelOfInsert`).

The residual-multiplicity factorization, the inserted-site analogue of `TNLean.PEPS.regionInsertedCoeff_eq_smul_edgeInsertedCoeff`, is now landed. The residual sum at a fixed local configuration η at v (`TNLean.PEPS.insertResidual`) is not the bare blocked weight of R: it carries (i) a consistency delta forcing μ and η to agree on the v-incident edges running to $(\mathrm{insert}\, v\, R)^c$ (these are S-boundary but not R-boundary), formalized as the vanishing of the residual on a mismatched outer v-incident edge (`TNLean.PEPS.insertResidual_eq_zero_of_inconsistent`); and (ii) the interior bond multiplicity of the S-boundary edges away from R, which are free in the residual sum but pinned by μ, the inserted-site overcounting multiplicity `TNLean.PEPS.insertOuterBondProd`. The residual refines the bridge-label blocked weight of R (`TNLean.PEPS.insertResidual_eq_filter_regionBlockedWeight`), and grouping the bridge-label blocked-weight sum by overwriting the free outer edges with μ collapses it to the multiplicity times the residual when μ and η are consistent (`TNLean.PEPS.regionBlockedWeight_bridge_eq_smul_insertResidual`, with the fiber cardinality `TNLean.PEPS.insertFiber_card`). Assembling the local-configuration grouping (`TNLean.PEPS.regionBlockedWeight_insert_eq_sum_localConfig`) with this collapse gives the inserted-site factorization (`TNLean.PEPS.insertOuterBondProd_smul_regionBlockedWeight_insert` and `TNLean.PEPS.insertOuterBondProd_smul_insertResidual_eq`): the inserted-site multiplicity times the blocked weight of S is $\sum_\eta A.\mathrm{component}\, v\, \eta\, (\sigma_v)$ against the bridge-label blocked weight of R on the consistent configurations. The multiplicity is bond-dimension data, hence identical for A and for the reindexed $\widetilde{B}$; it therefore cancels in the quotient of the two region proportionalities, leaving $\sum_\eta A.\mathrm{component}\, v\, \eta \cdot (c_R\, \widetilde{B}$-residual).

The scalar extraction on top of this factorization is now landed. Assembling the inserted-site factorization with the residual collapse reads the inserted-site multiplicity times the blocked weight of insert $v\, R$ as the sum over the inserted-site local configurations of the inserted-site tensor against the bridge-label blocked weight of R on the configurations consistent with μ, and zero otherwise (`TNLean.PEPS.insertOuterBondProd_smul_regionBlockedWeight_insert_eq_bridge`). A consistent local configuration at v is determined by μ and its bridge label — it reads the bridge label on the v-incident edges bounding R and μ on the remaining v-incident edges — so two consistent configurations with the same bridge label coincide (`TNLean.PEPS.localConfig_eq_of_insertConsistent`). Feeding the two region proportionalities $A_R = c_R\, \widetilde{B}_R$ and $A_S = c_S\, \widetilde{B}_S$ (the outputs of `TNLean.PEPS.regionComplement_comparison` at R and $S = \mathrm{insert}\, v\, R$) through this factorization, cancelling the bond-only inserted-site multiplicity, and substituting the R-proportionality leaves a vanishing combination of $\widetilde{B}$'s R-blocked family weighted by $c_R\, A_v(\eta) - c_S\, \widetilde{B}_v(\eta)$ over the consistent η. The consistent configurations inject into their bridge labels, so linear independence of $\widetilde{B}$'s R-blocked family separates the combination to a single term each, forcing the per-vertex relation $A_v = (c_S/c_R) \cdot \widetilde{B}_v$ at every local configuration (`TNLean.PEPS.component_eq_of_regionProportional`). When the comparison tensor is the gauge-absorbed second tensor applyGauge $B\, X$, the reindexed inserted-site tensor is the gauge action gaugeVertex $B\, X\, v$, so the extraction reads as the per-vertex gauge relation $A_v = (c_S/c_R) \cdot \mathrm{gaugeVertex}\, B\, X\, v$ (`TNLean.PEPS.component_eq_gaugeVertex_of_twoBlockProportional`), exactly the shape of `hPV`. Consequently the torus scalar condition follows directly from the two two-block proportionalities at every vertex with a common ratio (`TNLean.PEPS.lambda_pow_card_torus_eq_one_of_twoBlockProportional`).

The production of a *single* per-edge family carrying the absorbed equality at every edge is now landed. The conjugation-to-absorbed conversion is packaged so the bare-edge absorbed equality at a single edge follows from one coefficient-identity witness, depending only on the gauge at that edge (the open-edge gauge cancellation isolates the boundary edge), so the per-edge absorbing choices are independent (`TNLean.PEPS.edgeAbsorbed_o `f_conjIdentity`, `TNLean.PEPS.edgeAbsorbed_of_edgeCoeffIdentityWitness`). Feeding the two orientation-class coefficient-identity witness families to that conversion produces one per-edge gauge family X — the orientation-adapted absorbing gauge read off each chosen witness — for which the bare-edge absorbed equality against applyGauge $B\,X$ holds at every torus edge (`TNLean.PEPS.exists_edgeAbsorbed_torus_rectangle`). The region-independence step then multiplies this back up to the region absorbed equality, supplying the comparison's `hregion` and producing the region-block proportionality $A_R \propto \widetilde{B}_R$ at *every* comparison region from the bare-edge equality (`TNLean.PEPS.regionAbsorbed_of_edgeAbsorbed`, `TNLean.PEPS.twoBlockProportional_of_edgeAbsorbed`); the proportionality scalar is unique once the comparison block is injective (`TNLean.PEPS.twoBlockScalarProportional_unique`). The torus scalar condition is then assembled directly from the bare-edge absorbed equality plus the per-vertex two-block proportionalities with a common ratio (`TNLean.PEPS.lambda_pow_card_torus_eq_one_of_edgeAbsorbed`).

What remains for an unconditional `hPV`, and hence for an unconditional torus Fundamental Theorem, are three pieces, all consuming the landed X above. First, the *orientation-uniformity* of that absorbing family — that it is one horizontal matrix and one vertical matrix up to per-edge scalars, the shape the Fundamental Theorem's gauge conclusion `TNLean.PEPS.IsTorusOrientationUniformGaugeFamilyModScalar` asserts. The absorbing matrix `TNLean.PEPS.absorbedBoundaryGauge` is Z^{T} or Z^{-1} according to whether the boundary edge's left endpoint lies in the red blocking region, and the engine gauge Z_e is the transported reference; both are orientation-determined, but the endpoint-membership branch varies across the translates of a class with wraparound, which is the transpose/orientation bookkeeping that remains. Second, the blocked-tensor injectivity of the comparison regions of the reindexed applyGauge $B\,X$ (the gauge-absorbed analogue of the rectangle injectivity of B), and of A's one-site-different comparison regions and their complements. Third, the translation-invariant common ratio c_S/c_R of the (now uniquely-named, `TNLean.PEPS.twoBlockScalarProportional_unique`) proportionality scalars across the vertices. The scalar extraction itself is no longer on the critical path.

6. Prove the scalar condition $\lambda^{nm} = 1$ for the square-lattice theorem, and state separately the scalar freedom in the non-translation invariant normal theorem.

   The square-lattice scalar condition is formalized: when all per-vertex scalars equal a single λ, their product over the width $\times$ height lattice is λ raised to the site count, which the injective theorem forces to be one, giving $\lambda^{\text{width}\cdot\text{height}} = 1$. The square-lattice theorem skeleton records the top-down assembly: from the orientation-uniform gauge absorption datum and the per-vertex single-scalar relation (the region comparison output) the scalar condition follows. A second entry to the same assembly consumes the gauge in the shape the edge blocking actually delivers: an orientation-uniform gauge fixed only up to vertex-balanced per-edge constants, which absorb identically, so the exact orientation-uniform reference carries the post-absorption equality and the per-vertex relation against it yields the same scalar condition. Both skeletons remain conditional on the open per-edge gauge of obligation 5.[52]

4 Source-to-formalization ledger

- Lemma `injective_union`: blueprint `lem:peps_injectiveRegion_union`, issue #1374. The positive-bond formalization is the sibling node `lem:peps_injectiveRegion_union_pos` (`TNLean.PEPS.regionBlockedTensorInjective_union_overlap`); the unrestricted node remains the source statement without the positive-bond hypothesis.
- Finite iteration of Lemma `injective_union`: blueprint `lem:peps_regionInjectivityUnionClosure_finite`, issue #1374.
- Regions R, S, and T: blueprint `def:peps_normalSquareBlockingRegions`, issue #1375. This node also records the conditional implication from coordinate rectangular injectivity, union closure, and a rectangular cover of the displayed T-region to the abstract R, S, T blocking-region structure.[53]
- Injectivity of the two edge blocks removed from the displayed T-window, and of their union under union closure: blueprint `lem:peps_normalSquareRegionT_edgeBlocks_injective`, issue #2004.
- General rectangular-cover criterion for square-lattice regions: blueprint `lem:peps_normalSquareRectangleCover_injective`, issue #2004.
- Rectangular-cover criterion for injectivity of the displayed T-region: blueprint `lem:peps_normalSquareRegionT_injective_of_cover`, issue #2004. This node also contains the rotated vertical local T-region cover criterion.
- Obstruction to applying that criterion directly to the present origin-based local-window T models and edge-complement models: blueprint `lem:peps_normalSquareRegionT_no_five_by_six_cover`, issue #2004.
- Finite-lattice complementary block around an edge: blueprint `lem:peps_normalSquareEdgeComplementRegion`, issue #2004.

[52] Formal declarations: `TNLean.PEPS.card_squareLatticeVertex`, `TNLean.PEPS.lambda_pow_card_eq_one`, `TNLean.PEPS.fundamentalTheorem_normalSquarePEPS`, `TNLean.PEPS.tiNormalGaugeAbsorptionData_of_modScalar`, and `TNLean.PEPS.fundamentalTheorem_normalSquarePEPS_of_modScalarGauge`.

[53] The formal declaration is `TNLean.PEPS.normalSquareBlockingRegions_of_TCover`.

- Rectangular-cover criterion for injectivity of that block: blueprint `lem:peps_normalSquareEdgeComplementRegion_injective_of_cover`, using the cover definition `def:peps_normalSquareRectangleCover`, issue #2004.

- Extension of a local T rectangular cover to a rectangular cover of the normalized horizontal-edge complementary block by adjoining the two top-collar rectangles: blueprint `lem:peps_normalSquareEdgeComplementRegion_coverOfT`, issue #2004.

- Normalized 5×7 top-collar decomposition of that block: blueprint `lem:peps_normalSquareEdgeComplementRegion_topCollar`, issue #2004. This node also contains a normalized 7×5 collar identity for the analogous rectangular frame and the rotated vertical edge-complement collar identity.

- Normalized horizontal and vertical red/blue/complement blocking data: blueprint `lem:peps_normalSquareHorizontalEdge_blockingData`, issue #2004. These use the common one-edge blocking-data structure. They also expose the corresponding injective-chain conclusions. The horizontal complement injectivity is reduced to a local T-cover, and the vertical complement injectivity is reduced to a rotated local T-cover.

- Translated horizontal red/blue/complement blocking data: blueprint `lem:peps_normalSquareHorizontalTranslatedEdge_blockingData`, issue #2004. This is the coordinate-origin-parametric horizontal step toward the every-edge construction. Its coordinate origin specializes to the normalized horizontal picture.

- Translated vertical red/blue/complement blocking data: blueprint `lem:peps_normalSquareVerticalTranslatedEdge_blockingData`, issue #2004. This is the coordinate-origin-parametric vertical step toward the every-edge construction. Its coordinate origin specializes to the normalized vertical picture.

- Conditional construction from translated edge windows: blueprint `thm:peps_normalSquareTranslatedEdgeBlockingHypotheses_of_windows`, issue #2004. This records the formal boundary before proving that every edge in the finite square lattice admits such a translated window. Coordinate right and upward edges are tied to the corresponding translated windows, and the margin hypotheses for actual coordinate right/up edges are explicit. Arbitrary horizontal and vertical edges inherit these criteria from their coordinate representatives. The oriented margin-and-cover data structure records the remaining per-edge input in the open rectangular coordinate model. One such datum directly supplies one-edge blocking data together with the corresponding endpoint, injectivity, disjointness, and coverage conclusions, and a family of such data determines the normal edge-blocking hypotheses. The resulting hypotheses recover the supplied datum and the same combined edge-blocking conclusions at each edge. The horizontal and vertical margin predicates are named, the coordinate right/up and oriented-edge window constructors accept those predicates directly, translated windows on coordinate right/up edges force those predicates, and the open 7×7 graph does not admit a family of these translated windows over all edges; four explicit boundary examples are recorded as a single conjunction. The per-edge margin-cover structure stores the same predicates, and the resulting open-boundary obstruction for horizontal edges without enough left or right margin and vertical edges without enough bottom or top margin is also formalized, including explicit 7×7 examples and their four-sided conjunction. Thus the present margin criterion is a sufficient interior criterion rather than the final every-edge construction. Consequently, the nonexistence of a family of this margin-cover data over all edges of the current open 7×7 graph is also formalized.

- A family of one-edge blocking data determining the uniform every-edge hypotheses, including the combined endpoint, injectivity, disjointness, and coverage assertion: blueprint `thm:peps_normalEdgeBlockingHypotheses_injectiveChain`, issue #2004.
- Edge blocking into red, blue, and complementary injective regions: blueprint `thm:peps_normalEdgeBlockingToInjectiveChain`, issue #1375.
- Application of Lemma `inj_isomorph` after normal blocking: downstream issues #1367, #1368, #1363, and #1364.
- Translationally invariant gauge absorption: blueprint `thm:peps_tiNormalGaugeAbsorption`, issue #1375.
- Final R and S comparison: blueprint `thm:peps_oneVertexComplementComparison`, issue #1360.
- Theorem 3: blueprint `thm:fundamentalTheorem_normalSquarePEPS`, issue #1375.
- General theorem `normal`: blueprint `thm:fundamentalTheorem_normalPEPS`, issue #1375. A scope-restricted connected-graph form is now proved; see the next section.

5 The general-graph assembly and its scope restrictions

The general theorem `normal` is now assembled on an arbitrary connected finite simple graph, in a scope-restricted form.[54] The route is the source's: the

[54] Formal declarations: `TNLean.PEPS.fundamentalTheorem_normalPEPS` (existence, blueprint `thm:fundamentalTheorem_normalPEPS_connected`) and `TNLean.PEPS.fundamentalTheorem_n`

blocking hypotheses produce a per-edge gauge from the coherent three-block frames at every edge, the gauges are absorbed into the second tensor, the one-site comparisons give a nonzero scalar λ_v at every site, the closed state forces $\prod_v \lambda_v = 1$, and the spanning-tree edge-scalar solve of the injective theorem absorbs the scalars into the gauges, leaving the scalar-free local gauge relation. The uniqueness clause is the per-edge conjugator determinacy: two families realizing the scalar-free relation are proportional at every edge.

The four comparison points with the source statement (lines 1576–1583 of `Papers/1804.04964/paper_normal.tex`) are the following.

1. *Shared blockings, injective for both tensors.* The formal hypotheses instantiate the blocking bundle at the conjunction predicate: each blocking region is injective for both tensors at once. This is the reading under which the source's final comparison contracts both tensors over the same regions; the source phrase "*they* can be blocked" is about both states.

2. *Single crossing edge.* The per-edge gauge lives on the distinguished edge only when that edge is the entire bond between the red and blue blocks; with several red-to-blue crossings the isomorphism lemma would produce a gauge on the collected bond, not one matrix per edge. The formalization takes the single-crossing condition as an explicit hypothesis, reading "blocked *around* every edge" as in the source's Theorem `3` picture, where the two blocks adjacent to the edge touch only along it. The recorded blocking bundle does not carry this condition as a field.

3. *Bond dimensions derived from the blocking hypotheses.* The equality of the two bond dimension assignments is no longer a hypothesis of the existence theorem.[55] The derivation is the region-level analogue of the injective theorem's edgewise argument. Around each edge, the two blocked chains have state coefficients equal to positive merge-collapse constants times the original state coefficients, hence proportional states. Multiplying one super-site of each chain by the other chain's constant restores an exact state equality while preserving the super-site injectivities and the bond dimensions. The blocked insertion-algebra equivalence on the distinguished coarse bond then forces equal sizes (lines 560–583 of `Papers/1804.04964/paper_normal.tex`, the remark that the algebra isomorphism makes the bond dimensions on the two sides agree), and under the single-crossing hypothesis the coarse bond carries the original bond dimension of the distinguished edge for each tensor. The per-edge uniqueness clause keeps the equality among its hypotheses, since the families of gauges it compares are typed by the identification of the two bond spaces that the existence theorem produces.

`ormalPEPS_gauge_unique` (uniqueness, blueprint `thm:fundamentalTheorem_normalPEPS_gauge_unique`).

[55] Formal declaration: `TNLean.PEPS.bondDim_eq_of_normalBlocking` (blueprint `thm:peps_bondDim_eq_of_normalBlocking`), consumed by `TNLean.PEPS.fundamentalTheorem_normalPEPS`.

4. *Positive bonds and connectivity.* These are the counterexample-backed corrections carried over from the injective theorem (`docs/paper-gaps/peps_injective_ft_section3_route.tex` and `docs/paper-gaps/peps_gaugeConsistency_connectivity_gap.tex`); the scalar absorption is exactly the step that requires connectivity, matching the source's remark that the proportionality constants can be incorporated into the gauges.

The second point is resolved by reading the source: the single-crossing hypothesis is the chain structure of the source's hypothesis "blocked into three partite injective MPS around every edge" (line 1579 of `Papers/1804.04964/paper_normal.tex`), not a narrowing. The source's blocking around an edge takes the two endpoints of the chosen edge as the first two parties of the chain, so the chosen edge is the bond between them: "Choose one edge of the PEPS and group together all vertices except the endpoints of the edge. This regrouped tensor together with the two endpoints of the edge forms a three-partite MPS" (lines 981–1009). The normal case enlarges the first two parties to injective regions while keeping the distinguished edge their only contact: in the square-lattice picture of the proof of Theorem `3` (lines 1475–1498) the red and blue blocks meet only along the distinguished edge, which reappears as the bond between the first two coarse sites of the blocked chain. The proof then reads the gauge of the isomorphism lemma as living on that edge of the original network ("This establishes a gauge transformation on the edge $(1, 5)$ of the original PEPS", line 1037; "giving a gauge transformation on every edge", line 1498), which presupposes that the bond between the first two parties is the edge's own virtual leg, and the general theorem requires exactly the blockings of that proof ("The above proof can be repeated for any PEPS as long as it is possible to block into injective regions as required by" the two lemmas, line 1574). Since any edge between the red and blue regions necessarily belongs to the red-to-blue bond of a three-partite chain, the distinguished edge being the entire bond is equivalent to it being the only red-to-blue crossing. The source nowhere blocks an edge into a collected multi-edge bond between the first two parties, and no per-edge factorization of a product-bond gauge appears in the proof of Theorem `3`. The hypothesis is therefore kept in the formal statements unchanged and is no longer counted as a scope restriction; the in-source markers and the blueprint entries `thm:fundamentalTheorem_normalPEPS_connected` and `thm:peps_bondDim_eq_of_normalBlocking` state the reading explicitly so that it can be audited against the source lines above.

The first point is also resolved by reading the source: the shared blockings are the source's hypothesis, not a narrowing. The theorem blocks both states by one blocking: "Suppose two normal PEPS generating the same state satisfy the following: *they* can be blocked into three partite injective MPS around every edge" (lines 1577–1579 of `Papers/1804.04964/paper_normal.tex`). The proof it generalizes chooses the regions once, geometrically: "Let us block the TN into three injective parts around an edge. This can be done with e.g. the following choice of regions" (line 1475), and feeds the two blocked chains to the isomorphism lemma, whose statement takes two three-site MPS over the same tripartition with all six blocks injective (lines 254–278, applied at line 1498). The insertion correspondence driving the comparison is stated bond by bond between the two networks blocked identically ("inserting a matrix Z on any bond of the first PEPS gives the same state as inserting the same matrix Z on the corresponding bond of the second PEPS", line 1519), and the final comparison contracts both tensors over the same one-site-different regions R and S ("Let us denote the tensor on region R as A_R ($\tilde{B}_R$) and on region S as A_S ($\tilde{B}_S$)", line 1544), through the comparison lemma, whose hypothesis makes all four blocked tensors injective (lines 1067–1093). A pair of unrelated blockings, one per tensor, would not feed either lemma: the isomorphism lemma compares the two chains over one tripartition, and the comparison lemma cancels the common blocked region against both sides. The conjunction predicate of the formal hypotheses, each region injective for both tensors at once, is therefore the source's hypothesis set, and the connected-graph entry `thm:fundamentalTheorem_normalPEPS_connected` no longer counts it as a restriction; the remaining additions against the source statement are the counterexample-backed positivity and connectivity corrections shared with the injective theorem.

The source's remark after the theorem (line 1584 of `Papers/1804.04964/paper_normal.tex`) states that for a translationally invariant system the gauges are also translationally invariant, provided the proportionality constants are not absorbed into the gauges. On the torus, the only translation-invariant geometry the formal development carries, this is already delivered: the absorbed gauge family of the unconditional torus theorem is translation covariant, the orientation-faithful form of "the same matrix X (Y) on all horizontal (vertical) edges" with the transposed inverse on the seam edges of the ordered edge convention (see the section on closure on the torus), and the theorem keeps the single scalar λ with $\lambda^{nm} = 1$ separate from the gauges, matching the parenthetical.[56] A form of the remark for translation-invariant settings other than the torus is not formalized: the development carries no other translation-invariance predicate, and none of its statements assert translationally invariant gauges outside the torus. This is recorded as a non-goal of the present route rather than a difference from the theorem: the remark follows the theorem instead of being part of its statement, and the connected-graph assembly takes the remark's other branch, absorbing the constants into the gauges.

Two degenerate one-site comparisons allowed by the hypothesis bundle are handled directly rather than through the boundary-edge engine: when the smaller comparison region is empty its block is the count of global virtual configurations, and when the one-site completion is the full vertex set its block is the closed state coefficient; in both cases the comparison proportionality holds with scalar one. Every other comparison region has a boundary edge by connectivity.

[56] Formal declarations: `TNLean.PEPS.exists_torusCovariantAbsorbedGaugeFamily` and `TNLean.PEPS.fundamentalTheorem_normalTorusPEPS_unconditional` (blueprint `thm:fundamentalTheorem_normalTorusPEPS_unconditional`).

6 Blueprint diagrams

The normal route has its own tensor-network diagrams. These are not substitutes for the injective theorem diagrams; they attach the additional normal blocking layer to named proof nodes.[57]

- Lemma `injective_union` shows the four regions $A \setminus B$, $A \cap B$, $B \setminus A$, and $(A \cup B)^c$.
- Definition `peps_normalSquareBlockingRegions` shows the regions R, S, and T from the square-lattice proof.
- Lemma `peps_normalSquareEdgeComplementRegion_topCollar` shows the finite-lattice complementary block A_3 as the local T-region together with the two top-collar rectangles in the normalized 5×7 frame.
- Theorem `peps_normalEdgeBlockingToInjectiveChain` shows the red/blue/complementary blocking around a distinguished edge and the resulting three-site chain.
- Theorem `peps_tiNormalGaugeAbsorption` shows the final translationally invariant local gauge formula with horizontal gauge X and vertical gauge Y.

7 Conclusion

The normal PEPS route is separated from the injective PEPS route in the blueprint and in the GitHub issue tree. The tensor-level proof of Lemma `injective_union` is now complete for positive bond dimensions, in both the disjoint and the overlapping case; the region-injectivity vocabulary, the four-region decomposition, and the abstract contraction-chain criterion that supported it are present. The next genuine mathematical steps are the source-faithful local and rotated T-injectivity argument and the every-edge finite-geometry construction. After these normal blocking hypotheses are proved, the downstream proof should reuse the injective edge-insertion route rather than duplicate it. The translationally invariant gauge absorption and the scalar condition remain the final steps of the normal theorem.

8 The vertex-frame obstruction and the block-frame route

The region insertion machinery formalized so far inserts the comparison matrix at the *in-region endpoint vertex* v of the boundary edge f and realizes it through

[57] Chapter 13a writes the diagrams as native bodies beside the corresponding statements. The web blueprint captures the same bodies and renders them through the generic tenkz bridge.

v's tensor (`regionInsertionOp`). With this frame the coefficient transfer reduces cleanly to a single residual-frame statement: the v-side row of the first tensor is of single-bond insertion form. Every step of that reduction uses only blocked-region injectivity of the region and its complement.

The remaining v-side row statement, however, does not close under blocked-region injectivity alone. Discharging it evaluates the first tensor's endpoint operator, which factors as $\Phi_v^A \circ (\text{incident insertion}) \circ \Psi_v^A$ with Ψ_v^A a left inverse of Φ_v^A (`localTensorMap` of the first tensor at v), on the *second* tensor's region weight vectors. This requires the cross-tensor endpoint image inclusion $\text{range}(\Phi_v^B) \subseteq \text{range}(\Phi_v^A)$, i.e. the two single-vertex tensor images at v coincide. That coincidence is available only from single-vertex spanning (`span_stateOpenCoeff_eq_top`, derived from `vertexComplementTensorInjective_of_isVertexInjective`), which is genuine *vertex* injectivity. Blocked-region injectivity is linear independence of the blocked weight family indexed by boundary configurations; for a region with more than one site it does not force the single-vertex marginal at v to be injective, and two tensors with the same global state can have distinct v-column spaces. The disanalogy with the edge argument is structural: the edge resonate identity is single-tensor (the abstract operators act on one family's components), whereas the region resonate identity reads the first tensor's operator on the second tensor's vectors, introducing a cross-tensor image mismatch absent at the edge level.

Hence the vertex-endpoint frame closes only the *vertex-injective* specialization, where the injective Fundamental Theorem already supplies the gauge (`fundamentalTheorem_normalPEPS_of_vertexInjective`). The source's actual argument is block-framed: the injective blocks play the role of super-sites, and the insertion is realized on the *block's* open boundary leg through the blocked tensor, so the image-preservation step becomes the block-level range coincidence range(blocked map of A) = range(blocked map of B), which is provable from blocked-region injectivity together with the same-state hypothesis (the blocked-weight analogue of `range_localTensorMap_eq_of_sameState`). The conditional chain from the coefficient transfer to the per-edge gauge is already block-injective-clean (`exists_regionEdgeGauge_of_coeffTransfer`, no single-vertex injectivity), so a block-framed proof of the coefficient transfer feeds it directly. The block-framed realization—a block-leg insertion operator with its own block-level resonate inversion—is the genuine remaining obligation for the general region-injective theorem and the recommended next development.

This block-framed realization is now delivered through the coarse three-site route. The three injective blocks are assembled into a coarse tensor on a three-vertex complete graph whose super-sites are injective directly from the blocked-region injectivities, with no single-vertex injectivity of the original tensor; the proven injective inserted-coefficient correspondence runs on its distinguished super-bond, the bond-model collapse descends the coarse coefficient to the single-edge region-inserted coefficient, and the shared positive fiber multiplicity cancels. The result is the block-framed coefficient transfer in both directions (`exists_regionInsertedCoeff_eq_sharedRegion`, `exists_regionInsertedCoeff_eq_sharedRegion_symm`), the two bond-local transfer kernels (`isBondLocalTransferKernel_of_coherentFrames`), the forward multiplicativity (`regionEdgeTransfer_mul`), and hence the per-edge gauge (`exists_regionEdgeGauge_of_coherentFrames`), for two coherent frames sharing the three regions, the bond dimensions, and the single distinguished crossing edge. The block-level step $V = W$ and the forward multiplicativity — the genuine remaining obligation above — are thereby closed modulo review, conditional only on the coherent coarse blocking frames the source's edge geometry supplies. What remains for the general theorem is constructing those frames at every edge from the source's finite-lattice blocking, the geometry obligation recorded earlier in this note.

Closure on the torus: the unconditional theorem

The geometry obligation is discharged on the discrete torus. The coherent frames are instantiated at every edge from the contiguous-rectangle injectivity hypotheses, and the per-edge gauges are absorbed into the second tensor, giving the bare-edge absorbed inserted-coefficient equality at every torus edge. Three further steps complete the assembly. First, a gauge preserves blocked-region linear independence (`regionBlockedTensorInjective_applyGauge`): the gauged blocked weights are the original weights mixed by an invertible boundary coupling, the product over the boundary edges of the surviving endpoint gauges. Second, the absorbing family is constructed covariantly: each edge carries the transported reference gauge of its orientation class, so the family commutes with the torus translations (`exists_torusCovariantAbsorbedGaugeFamily`). Third, translation covariance transports the corner-region comparison to every site with the same proportionality scalars (`twoBlockScalarProportional_translate`), so the inserted-site extraction yields one ratio $\lambda = c_S/c_R$ at every vertex. The result is the unconditional torus theorem (`fundamentalTheorem_normalTorusPEPS_unconditional`): from translation invariance, matched positive bond dimensions, the same state, and the rectangle injectivity hypotheses alone, there are a translation-covariant gauge family and a single scalar λ with $A_v = \lambda \cdot (\text{gauge action of } B)_v$ at every site and $\lambda^{nm} = 1$. The union closure of injective regions is derived from the positive bond dimensions, not assumed.

One convention finding accompanies the closure. The ordered-edge normal form stores a wraparound edge with its endpoints swapped, so the absorbing family carries the transposed inverse of the class matrix on the seam edges. A family that is one matrix per orientation class up to scalars in the stored convention cannot satisfy the absorbed equality at the seam (it would force Z^{T} proportional to Z^{-1}), so the lexicographic uniform-up-to-scalar predicate misstates the source's "same matrix on all horizontal (vertical) edges" on the torus; translation covariance is the orientation-faithful form. The earlier conditional torus statement, whose per-vertex hypothesis is phrased against a lexicographically uniform family, is unsatisfiable at seam edges in general and is superseded.

The superseded apparatus has been removed. Gone are: the conditional torus statement itself (`fundamentalTheorem_normalTorusPEPS`); the lexicographic uniform-up-to-scalar predicate and its selection lemmas, which gathered per-edge agreement with one matrix per orientation class into a uniform family (`IsTorusOrientationUniformGaugeFamilyModScalar`, `TorusOrientationGauge.lean`, `TorusTINormalGauge.lean`, `TorusClassAgreement.lean`); the production of such a family from the rectangle hypotheses and its per-edge feeders (`isTorusOrientationUniformGaugeFamilyModScalar_torus_rectangle`, `isTorusOrientationUniformGaugeFamilyModScalar_of_edgeWitnesses`, `edgeCoeffIdentityWitness_translateUniform`, `TorusRectangleWitness.lean`); and the per-edge-chosen absorbed family superseded by the covariant construction (`exists_edgeAbsorbed_torus_rectangle`). The witness interface, the witness transport along translations, the per-class witness families, the single-edge absorbed-equality conversion, and the determinacy bridges all remain: they carry the covariant route of the unconditional theorem.

The uniqueness clause of Theorem 3 — "X and Y are unique up to a multiplicative constant" — is closed. Two families realizing the bare-edge absorbed equality at an edge are proportional there (`torusAbsorbedGauge_unique_scalar`): each family's absorbed equality lifts to the region inserted-coefficient identity in conjugation form (`regionConjIdentity_of_edgeAbsorbed`), the transfer map is determined by the identity, and the centralizer of the bond matrix algebra is the scalars. For translation-covariant families the per-edge scalar is one constant per orientation class, transported along the translations (`torusCovariantAbsorbedGauge_unique_classScalar`); the honest class statement carries the constant c on the edges stored in the natural cyclic order and c^{-1} on the seam edges, because the covariance places the transposed inverse of the class matrix on a seam edge and the transposed inverse of a proportionality inverts its constant. The clause also holds against the per-vertex relation itself (`torusGauge_unique_scalar_of_perVertex`): the inserted-edge coefficient is multilinear in the site tensors, so the relation $A_v = \lambda \cdot (\text{gauge action of } B)_v$ with $\lambda^{nm} = 1$ already implies the bare-edge absorbed equality at every edge (`edgeAbsorbed_of_perVertex`).

What remains against the source statement of Theorem 3: the source's square-lattice region form of the theorem is covered separately by the conditional square-lattice statement recorded earlier in this note. The torus sizes match the source: the theorem and its uniqueness clauses hold for $n, m \geq 7$ (see the next section).

The torus sizes: from $n, m \geq 9$ to the source's $n, m \geq 7$

An earlier formal statement required $n, m \geq 9$. The margin audit traced each size hypothesis to its consumer. The blocking around an edge removes a 2×3 block and a 3×2 block whose union has a 5×4 bounding box, and the complementary region is decomposed into wraparound-free rectangular bands;

a band of width one is not a union of 2×3 and 3×2 rectangles, so each side margin of the bounding box must be zero or at least two, and the side where the filler rectangle dips below the removed blocks must be at least two. The earlier placement put the bounding box in the interior with margins of at least two on all four sides, forcing $5+2+2 = 9$ in the wide direction. The crossing-uniqueness window used the strict bounds only to rule out wrapped cyclic steps.

The repaired placement anchors the bounding box against the far seam: margins of at least two below and to the left, margin exactly zero above and to the right (anchor $(n-5, m-5)$, hypothesis shape $x + 5 = n \vee x + 7 \leq n$ per direction). The seam-side bands are empty and drop out of the cover, and a cyclic step wrapping the seam lands in the zero column or row, strictly outside both removed blocks, so crossing uniqueness survives. This packs both reference blockings and the 3×3 corner comparison into every torus with $n, m \geq 7$, the source's sizes.

The source's own packing is different: the example regions in the proof of Theorem 3 (lines 1475–1498 of `paper_normal.tex`) sit on a 7×7 patch with margins of width one on three sides, and the complementary region is read as connected through the torus seam — its injectivity uses 2×3 and 3×2 regions that wrap. That reading is licensed by the source's hypothesis, which makes *every* region of those shapes injective, wrapped translates included. The formal hypothesis (`NormalTorusRectangleInjectivityHypotheses`) is weaker: it covers only the wraparound-free coordinate rectangles. The seam-anchored placement reaches the source's sizes from this weaker hypothesis, so no paper gap arises, and the source's $n, m \geq 7$ is justified under either reading.

Removal of the conditional square-lattice skeleton

The conditional square-lattice skeleton recorded earlier in this note has been removed. Its assembly assumed single-site injectivity of A (`IsVertexInjective`), the hypothesis the normal theorem exists to avoid: its scalar condition rested on the vertex-injective product identity rather than the region-injective one (`prod_perVertexScalar_eq_one_of_regionInjective`), and the interior-window open-lattice theorem (`fundamentalTheorem_normalSquarePEPS_unconditional`) and the general-graph gauge-equivalence theorem (`fundamentalTheorem_normalPEPS`) now cover the square-lattice content from region injectivity alone. Gone are: the conditional statements themselves (`fundamentalTheorem_normalSquarePEPS`, `fundamentalTheorem_normalSquarePEPS_of_modScalarGauge`, `lambda_pow_card_eq_one`, `card_squareLatticeVertex`); the orientation-uniform carrier on the open lattice (`SquareLatticeUniformBondDim`, `orientationUniformGauge`, `IsOrientationUniformGaugeFamily`, `IsOrientationUniformGaugeFamilyModScalar`, their selection lemmas, and `NormalSquareTI.lean`); and the translation-invariant absorption datum and its feeders (`tiAbsorb`, `TINormalGaugeAbsorptionData`, `tiNormalGaugeAbsorption`, `tiNormalGaugeAbsorptionData_of_modScalar`, `exists_postAbsorption_of_vertexInjective`, `gaugeEquivModEdgeScalars_of_matrixScalar`, absorbEdgeGauges_eq_of_matrixScalar, and NormalSquareTIAbsorption .lean). The vertex-injective product identity prod_perVertexScalar_eq_one itself remains: it serves the injective theorem (Theorem 2), where single-site injectivity is the source hypothesis. A source-exact translationally invariant square-lattice statement awaits a translation-invariance predicate on the open lattice, as the earlier sections record.

The closed-chain MPS corollary

The first corollary after the theorem normal (lines 1585–1622 of Papers/1804. 04964/paper_normal.tex) specializes it to one dimension: two normal MPS $\{A_i\}$, $\{B_i\}$ on $n \geq 3L$ sites such that blocking any L consecutive sites gives an injective tensor, generating the same state, are related by invertible matrices Z_i (one per bond, $n+1 \equiv 1$) with $B_i = Z_i^{-1} A_i Z_{i+1}$, unique up to a multiplicative constant. This corollary is now formalized by instantiating the connected-graph theorem on the cycle graph.[58] The closed chain is the cycle graph on n vertices, a block of L consecutive sites is a cyclic arc, and the blocking hypotheses of the general theorem are discharged by arc arithmetic: around the edge joining a site to its cyclic successor the chain splits into the L sites ending at the first endpoint, the L sites starting at the second, and the remaining $n - 2L \geq L$ sites; at every site the $L+1$ sites starting there and the L sites starting at its successor differ exactly at the site; arcs of length between L and n are unions of length-L arcs, hence injective by the union lemma; and the single-crossing hypothesis holds because the far ends of the two blocks around an edge are separated by the remaining $n-2L \geq L$ sites, so the only nearest-neighbour pair joining them is the edge itself. The hypotheses beyond the source's wording are those of the connected-graph theorem (positive physical and bond dimensions, recorded in the comparison points above), plus $0 < L$, which is implicit in the source's "blocking any L consecutive sites"; connectivity of the cycle is proved, not assumed, and the equality of bond dimensions is derived.

Two relationships are worth recording. First, the corollary is a different theorem from the Fundamental Theorem of the development's MPS chapters (TNLean/MPS/FundamentalTheorem/, after arXiv:1606.00608): those treat one site-independent tensor whose matrix-product traces agree at *every* chain length (SameMPV) and produce one global similarity, whereas the corollary fixes one chain length n, allows site-dependent tensors, and produces one gauge per bond. A dictionary between site-independent MPSTensor families and cycle-graph tensors (sending a single tensor to its repetition around the cycle and matching the trace coefficients with the closed-state coefficients) would let the two settings be compared; it is not small, because the cycle-graph tensor indexes

[58] Formal declarations: TNLean.PEPS.fundamentalTheorem_normalMPS_cycle (existence, blueprint cor:fundamentalTheorem_normalMPS_cycle) and TNLean.PEPS.fundamentalTheorem_normalMPS_cycle_gauge_unique (uniqueness, blueprint cor:fundamentalTheorem_normalMPS_cycle_gauge_unique), with the blocking construction in TNLean/PEPS/CycleArcRegion.lean and TNLean/PEPS/CycleBlockingData.lean (blueprint thm:peps_cycleNormalPEPSBlockingHypotheses).

its virtual legs by incident edges through the linear order on the vertices, and remains a follow-up. Second, the source's translationally invariant corollary (lines 1624–1661, single gauge Z and a root of unity λ with $\lambda^n = 1$) is not delivered by this instantiation: it needs a translation apparatus on the cycle mirroring the torus one (`TNLean/PEPS/TorusTranslation.lean`), and the development's translation-invariant setting remains the torus (`fundamentalTheorem_normalTorusPEPS_unconditional`). The strengthening of the corollary to $n \geq 2L+1$ announced by the source (line 1623, proved in its Section `normal_alt`) is likewise not formalized.

The matrix-level closed-chain corollary

The dictionary between site-independent matrix tensors and cycle-graph tensors, recorded above as a follow-up, is now in place for one repeated tensor, and the closed-chain corollary has a matrix-level form. The cycle tensor of a matrix tensor places one copy of the matrix family at every site of the closed chain (`cycleTensorOfMPS`), built on the bijection between sites and bonds (`cycleSuccEdge`) and the two incident bonds of each site. The state bridge identifies the graph-level state coefficient with the closed-chain matrix-product trace (`stateCoeff_cycleTensorOfMPS`), through a path expansion of word products (`MPSTensor.evalWord_ofFn_apply`) and its cyclic closure (`MPSTensor.trace_evalWord_eq_sum_cyclic`). The injectivity bridge identifies the blocked tensor of an arc of L consecutive sites with the length-L word products carrying the two pinned boundary bonds, each free bond away from the arc contributing a factor of the bond dimension (`regionBlockedWeight_cycleTensorOfMPS`); linear independence of the blocked family over the boundary pairs is then the nondegeneracy of the trace pairing against the spanning word products, so L-block injectivity of the matrix tensor gives arc injectivity of its cycle tensor (`regionBlockedTensorInjective_cycleTensorOfMPS`).

Feeding the bridges into the closed-chain corollary and unpacking the per-edge gauges yields the matrix form (`fundamentalTheorem_normalMPS`, blueprint `cor:fundamentalTheorem_normalMPS`): two L-block-injective matrix tensors whose matrix product vectors agree at size n are related by invertible per-bond matrices Z_v with $B^i = Z_v^{-1} A^i Z_{v+1}$, and two such families differ by a single constant (`fundamentalTheorem_normalMPS_gauge_unique`, blueprint `cor:fundamentalTheorem_normalMPS_gauge_unique`). The existence clause now carries the optimal $n \geq 2L+1$ of the source's alternative proof: `fundamentalTheorem_normalMPS` delegates to the overlapping-window corollary `fundamentalTheorem_normalMPS_of_overlap` (documented in the section "The strengthening to $n \geq 2L+1$ via overlapping windows" below). The uniqueness clause carries the same $n \geq 2L+1$: the source's uniqueness needs no system size of its own, and the argument that delivers it needs nothing beyond the size that produced the two gauge families. The unpacking meets the same seam convention as on the torus: the ordered-edge normal form stores the wraparound edge with its endpoints swapped, so the per-bond gauge is the transpose of the per-edge matrix away from the seam and its inverse on the seam edge (`cycleGaugeOfEdgeGauge`); the per-site component identity is equivalent to the conjugation relation through the two-bond factorization of the gauge action (`gaugeVertex_cycleTensorOfMPS`, `cycleGauge_component_iff_matrix`). The single-constant uniqueness compares the two families directly and then merges: iterating both conjugation relations along the spanning length-L word products of A makes the transports from a bond to its L-th successor agree on the whole matrix algebra, so the two gauges at every bond differ by a scalar (`gaugeFamily_bond_proportional`, blueprint `lem:gaugeFamily_bond_proportional`), and the conjugation relation for both families fixes the nonzero tensor B by the scalar $\mu_v^{-1}\mu_{v+1}$, so consecutive per-bond constants agree around the cycle. The matrix and site-dependent closed-chain statements now assume positive bond dimension, matching the source's tensor-network setting and excluding the vacuous 0×0 matrix algebra; a vanishing physical dimension is already excluded by block injectivity. The statements specialize the source's site-dependent families to one repeated tensor; the source's translation-invariant corollary (lines 1624–1661: a single gauge Z and a constant λ with $\lambda^n = 1$) is delivered in single-gauge form in the next section, and the strengthening of the existence clause to $n \geq 2L+1$ (line 1623, proved in the source's Section `normal_alt`) is now in place via the overlapping-window route below.

The translation-invariant corollary in single-gauge form

The translation-invariant corollary (lines 1624–1661: a single gauge Z and a constant λ with $\lambda^n = 1$ such that $B = \lambda \cdot Z^{-1}AZ$, with Z unique up to a multiplicative constant) is now delivered (`fundamentalTheorem_normalMPS_translationInvariant`, `fundamentalTheorem_normalMPS_translationInvariant_gauge_unique`, blueprint `cor:fundamentalTheorem_normalMPS_translationInvariant`, `cor:fundamentalTheorem_normalMPS_translationInvariant_gauge_unique`). The existence clause carries the optimal $n \geq 2L+1$ of the source's alternative proof: it delegates to the overlapping-window single-gauge corollary `fundamentalTheorem_normalMPS_translationInvariant_of_overlap` (documented in the section "The strengthening to $n \geq 2L+1$ via overlapping windows" below). The same collapse of the per-bond family is available size-independently — iterating the per-bond relation $B^i = Z_v^{-1}A^iZ_{v+1}$ along a word conjugates the word products of B by the gauges at the two ends, consecutive gauges are proportional by the centralizer step (`gaugeFamily_succ_proportional`), the same-state relation pins consecutive scalars against the nonzero tensor B, and one circuit of the closed chain forces $\lambda^n = 1$. The uniqueness clause needs no system size at all: two single-gauge realizations, iterated along the spanning words (`evalWord_eq_smul_conj_of_gauge`), give two equal conjugations of the full matrix algebra after the empty word pins

$\lambda^L = \lambda'^L$, so the gauges are proportional; neither equality of states nor the root-of-unity conditions enter.

The Applications translation-invariant description corollary

The Applications section contains a different translation-invariance corollary (lines 1801–1890 of `Papers/1804.04964/paper_normal.tex`), tracked by GitHub issue #2920. Its input is a single site-dependent injective MPS whose resulting closed-chain state is invariant under the cyclic shift. Its conclusion is not a gauge between two given tensor families: the proof applies the non-translation invariant closed-chain theorem to the family and its cyclic shift, obtains per-bond gauges Z_i, and then telescopes them to construct a repeated injective tensor $B = A_1 Z_1^{-1}$ that gives the same state. Writing T for the cyclic shift, the source assertion is the implication

$$\begin{gathered} T\Psi(A_1, \ldots, A_n) = \Psi(A_1, \ldots, A_n) \\ \Longrightarrow \\ D_1 = \cdots = D_n \quad \text{and} \quad \exists B,\ B \text{ is injective and } \Psi(A_1, \ldots, A_n) = \Psi(B, \ldots, B), \end{gathered}$$

with $B = A_1 Z_1^{-1}$ after the gauges from the comparison with the shifted chain are telescoped.

This corollary is not covered by the two-tensor results above. The already formalized statements compare two supplied tensors, or two supplied repeated tensors, with the same closed-chain state. They do not start from one site-dependent family and a cyclic-shift invariance hypothesis on its state, and they do not construct the comparison tensor internally. A uniform matrix-tensor version would be faithful to the existence of the repeated description, but would make the source's "all bond dimensions are the same" conclusion vacuous: the type already fixes one bond dimension D. The fully faithful statement therefore requires a per-bond, rectangular matrix model in which the adjacent invertible gauges first imply equality of the bond dimensions. Until that vocabulary exists, the intended first theorem is the uniform-D existence statement, carrying this scope restriction explicitly; the per-bond-dimension theorem remains a separate follow-up. In symbols, the initial restricted statement is only

$$T\Psi(A_1, \ldots, A_n) = \Psi(A_1, \ldots, A_n) \quad \Longrightarrow \quad \exists B,\ B \text{ is injective and } \Psi(A_1, \ldots, A_n) = \Psi(B, \ldots, B),$$

under a prior uniform bond-dimension hypothesis on the family A_i. The blueprint records this restricted target as `cor:exists_constant_injectiveMPS_of_cyclicShiftInvariantState`, now formalized as `TNLean.PEPS.exists_constant_injectiveMPS_of_cyclicShiftInvariantState`. The cyclic-shift comparison, the running-product (L_i, R_i) expression, the seam-closure argument pinning the loop product $P_n = Z_{n-1} \cdots Z_0$ to a scalar λ, and the final repeated-tensor construction $B = c\,(A_0 Z_0)$ with $c^n = \lambda^{-1}$ are all in place.

The n-th root absorbs the residual scalar that the source elides when it asserts $R_{n+1} = L_{n+1} = 1$: the seam closure forces P_n to be central, hence a scalar, but not necessarily the identity, so an n-th root (available since $\mathbb{C}$ is algebraically closed) is needed to fold the residual phase into the repeated tensor.

Verdict: scope restriction. The uniform-D theorem records the existence of the repeated injective tensor B, but it cannot express the source's equal-bond-dimension conclusion as a theorem with mathematical content: the type `MPSChainTensordDn` already fixes one bond dimension. The per-bond rectangular version, in which adjacent invertible gauges first imply equality of the bond dimensions, remains the follow-up.

The 2D analogue

The Applications section states the same corollary in two dimensions immediately after the one-dimensional version (lines 1892–1894 of `Papers/1804.04964/paper_normal.tex`), tracked by GitHub issue #2921. Its input is an injective 2D PEPS whose generated state is invariant under the lattice translations; its conclusion is that all vertical bond dimensions agree, all horizontal bond dimensions agree, and the state has a translation-invariant 2D PEPS description with a single injective tensor B of the same horizontal and vertical bond dimensions. The source states this corollary without proof (lines 1892–1894); it gives a proof only for the one-dimensional case (lines 1807–1890) and remarks at line 1801 that "below we provide the proof for injective MPS, but the proof can easily be extended to the other cases as well." The intended extension applies the one-dimensional argument separately along the horizontal and the vertical lattice translations: comparing the tensor with each translate through the injective PEPS Fundamental Theorem would produce per-edge gauges, which telescope around each row and each column and close under the periodic boundary condition. This route is the expected one, not a proof the source carries out.

The load-bearing dependency, the injective PEPS Fundamental Theorem (`TNLean.PEPS.fundamentalTheorem_PEPS`, blueprint `thm:fundamentalTheorem_PEPS`), is formalized. The one-dimensional repeated-tensor construction discussed above is also formalized as `TNLean.PEPS.exists_constant_injectiveMPS_of_cyclicShiftInvariantState`. The remaining work is the PEPS-specific two-directional closure: the horizontal and vertical gauge telescopings must be made compatible around plaquettes and periodic seams, and their residual virtual operations must be absorbed into one site-independent 2D tensor. In the notation of the blueprint proof sketch, writing

$$H_{j,k} = R_h^{(j,k)}\big(L_h^{(j+1,k)}\big)^{-1}, \qquad V_{j,k} = R_v^{(j,k)}\big(L_v^{(j,k+1)}\big)^{-1},$$

the residual $H_{j,k}$ acts on the horizontal virtual leg $\mathbb{C}^{D_h}$, while $V_{j,k}$ acts on the vertical virtual leg $\mathbb{C}^{D_v}$. Thus the plaquette condition is not an ordinary product of a $D_h \times D_h$ matrix with a $D_v \times D_v$ matrix. It must be stated on the corner virtual space $\mathbb{C}^{D_h} \otimes \mathbb{C}^{D_v}$: after the row and column telescopings make

$H_{j,k} = H_k$ and $V_{j,k} = V_j$, the missing scalar plaquette flatness relation is

$$(H_k \otimes I_{D_v})(I_{D_h} \otimes V_{j+1}) = \lambda_{j,k}\,(I_{D_h} \otimes V_j)(H_{k+1} \otimes I_{D_v})$$

for nonzero scalars $\lambda_{j,k}$. These scalars must also satisfy the periodic seam closures

$$\prod_{k \in \mathbb{Z}/N_v\mathbb{Z}} \lambda_{j,k} = 1, \qquad \prod_{j \in \mathbb{Z}/N_h\mathbb{Z}} \lambda_{j,k} = 1.$$

The blueprint records the restricted target as `cor:exists_constant_injectivePEPS_of_translationInvariantState`, marked not ready.

Verdict: scope restriction. As in the one-dimensional case, the orientation-uniform torus bond-dimension type fixes the horizontal dimension D_h and the vertical dimension D_v in advance, so the source's "all vertical (resp. horizontal) bond dimensions are equal" clauses are vacuous in the uniform-dimension statement. A fully faithful version requires a per-edge, rectangular bond model in which the adjacent invertible gauges along each orientation first force equality of the bond dimensions; that variant inherits the same uniform-dimension restriction recorded for the one-dimensional corollary and remains a separate follow-up.

The strengthening to $n \geq 2L + 1$ via overlapping windows

The plan recorded in the previous section is carried out: the source's Lemma 5 (lines 2045–2255) is formalized for matrix tensors, and the closed-chain corollaries now hold at the source's bound $n \geq 2L+1$. The development stays at the matrix-tensor level throughout; no graph-level input beyond Skolem–Noether enters.

Four layers deliver the result. First, the linear-algebraic inputs: block injectivity extends to every longer block (`MPSTensor.isNBlkInjective_of_le`, the chain form of the source's remark at line 1940), and a word transport carries the length-k word products of one tensor to those of the other whenever the two closed-chain states agree at one length $n = k + q$ with both window lengths spanning (`MPSTensor.exists_mpvTransport`; the variant `MPSTensor.exists_evalWordTransport` starts from equal word products instead of equal traces). The transport is built by the left-inverse construction of the single-block linear extension: the pairing against the complementary word products is injective on one side by spanning and trace nondegeneracy, and the same-state hypothesis matches the two pairings on the window family.

Second, Lemma 5 itself (`MPSTensor.overlapWindow_exists_bondOperator`, uniqueness `MPSTensor.bondOperator_unique`, blueprint `lem:overlapWindow_exists_bondOperator`): deformed window tensors $C_0, \ldots, C_L$ on the $L+1$ overlapping windows around a bond that generate one common state on $n \geq 2L+1$ sites come from a single virtual operation X on the bond, with $C_0(a) = B^a X$ and $C_L(b) = X B^b$. The proof follows the source: comparing consecutive windows through the complementary arc of $n - L - 1 \geq L$ sites strips the state equality to the letter level (the inversions of lines 2140–2167), chaining the L letter relations along a window of $2L$ sites moves the deformation across the bond (the plugging of lines 2168–2210), and a combination of window products representing the identity extracts X from the two ends (lines 2211–2255, after the model of `eq:inj_O->X_argument`). The deformed window tensors play the role of the source's physical operators contracted with the blocked windows they act on; the statement is specialized to one site-independent tensor, matching the landed corollary surface, while the source states it for site-dependent tensors on five sites with $L = 2$.

Third, the insertion correspondence (`MPSTensor.exists_conjugation_of_mpv_eq`): for two L-block-injective tensors with the same state at one length $n \geq 2L + 1$, a virtual insertion X on the first network, realized on the overlapping windows and transported to the second, is extracted there as a bond operator $\Phi(X)$. Uniqueness of the bond operator makes Φ linear, unital and multiplicative — the source's algebra-homomorphism clause for $O_1 \mapsto X$ and $O_3^T \mapsto X$ (line 2253) — hence an automorphism of the matrix algebra, inner by Skolem–Noether; pairing insertions against all closed-chain words turns the conjugation of insertions into the conjugation of word products at length n, $B^w = ZA^wZ^{-1}$.

Fourth, a rigidity step the source leaves implicit in "similar to the injective case, this leads to" (line 2256): two L-block-injective tensors with *equal* word products at one length $n \geq 2L + 1$ are proportional letterwise by an n-th root of unity (`MPSTensor.proportional_of_evalWord_eq`). The transports of the identity at the window length and at the complementary length $n - 1 - L$ commute with every letter — their two one-letter extensions agree — hence are nonzero scalars, and peeling one letter off the full-length equality through the two spanning windows leaves $C^i = \lambda A^i$; both window lengths fit inside n exactly when $n \geq 2L + 1$, which is where the source's bound enters.

The capstones combine the layers: the translationally invariant corollary at $n \geq 2L + 1$ in single-gauge form, $B^i = \lambda Z^{-1} A^i Z$ with $\lambda^n = 1$ (`TNLean.PEPS.fundamentalTheorem_normalMPS_translationInvariant_of_overlap`, blueprint `cor:fundamentalTheorem_normalMPS_translationInvariant_of_overlap`), and the per-bond family $B^i = Z_v^{-1} A^i Z_{v+1}$ at $n \geq 2L + 1$ (`TNLean.PEPS.fundamentalTheorem_normalMPS_of_overlap`, blueprint `cor:fundamentalTheorem_normalMPS_of_overlap`), obtained by dressing the single gauge with powers of the root of unity, $Z_v = \lambda^v Z$. The single-gauge form arrives directly, so the matrix-level collapse of the per-bond family recorded in the previous sections is not needed on this route. The named matrix-level existence corollaries now carry this optimal bound: `TNLean.PEPS.fundamentalTheorem_normalMPS` and `TNLean.PEPS.fundamentalTheorem_normalMPS_translationInvariant` delegate to `fundamentalTheorem_normalMPS_of_overlap` and `fundamentalTheorem_normalMPS_translationInvariant_of_overlap`, so their hypothesis is $n \geq 2L + 1$ rather than the $n \geq 3L$ of the blocking route. The graph-level closed-chain corollary `TNLean.PEPS.fundamentalTheorem_normalMPS_cycle` keeps the $n \geq 3L$ of its three-disjoint-arc cover, which the overlapping-window apparatus replaces rather than reuses; the matrix-level existence clauses no longer route through it. The per-bond uniqueness clause `TNLean.PEPS.fundamentalTheorem_normalMPS_gauge_unique` no longer invokes the graph-level per-edge uniqueness: comparing the two gauge families along the spanning length-L words proves them proportional at every bond (`TNLean.PEPS.gaugeFamily_bond_proportional`) with no condition on the system size, so the clause carries the same $n \geq 2L+1$ as the existence clause that produces the families. The translation-invariant single-gauge uniqueness clause of the source corollary needs no system size and was already delivered by `TNLean.PEPS.fundamentalTheorem_normalMPS_translationInvariant_gauge_unique`. With this, every numbered claim of the source's Section `normal_alt` for matrix-product states is formalized for one site-independent tensor. The $2L+1$ strengthening for two-dimensional tensor networks (the corollary at lines 2297–2318) remains open.

The site-dependent form of Lemma 5

The site-dependent form of Lemma 5 is formalized, and the site-independent statements landed there are its constant specializations. The development works with the existing site-dependent chains (`MPSChainTensor`, one tensor per site of the closed chain, `TNLean/MPS/Chain/Defs.lean`) and adds the arc vocabulary: the ordered product of a word along an arc of consecutive sites, starting anywhere on the chain and wrapping cyclically (`MPSChainTensor.arcEval`), and the source's hypothesis that every window of L consecutive sites is injective (`MPSChainTensor.IsWindowInjective`, the general-L form of "the blocking of any two consecutive tensors is injective", lines 1928–1940), with the spanning of all longer arcs derived from it (`MPSChainTensor.IsWindowInjective.arc_span`, blueprint `def:chain_arc_window_injective`, `lem:isWindowInjective_arc_span`).

Lemma 5 itself is restated and proved for site-dependent chains (`MPSChainTensor.overlapWindow_exists_bondOperator`, blueprint `lem:chain_overlapWindow_exists_bondOperator`): deformed window tensors on the $L+1$ overlapping windows around a bond of a window-injective chain that generate one common state on $n \geq 2L+1$ sites come from a single virtual operation X on the bond, with the factorizations through the windows on the two sides and uniqueness of X (`MPSChainTensor.eq_of_span_mul_left`). The proof is the source's, with one bookkeeping change against the site-independent version: the letter-level stripping, the window chaining, and the two-ends extraction all track the starting site of every arc, and the two spanning families at the two sides of the bond are now distinct. The previously landed site-independent theorems (`MPSTensor.overlapWindow_exists_bondOperator`, `MPSTensor.bondOperator_unique`, `MPSTensor.exists_bondOperator_of_intertwine`) now delegate to the site-dependent ones through the constant chain (`MPSChainTensor.isWindowInjective_const`, `MPSChainTensor.arcEval_const`); their statements and the downstream corollaries are unchanged.

The algebra-homomorphism clause of the source lemma (line 2253) is delivered in the site-dependent setting by the insertion correspondence: one arc transport per window start (`MPSChainTensor.exists_arcTransport`; the rotated reading of the closed-chain state at every starting site is `MPSChainTensor.SameState.trace_arcEval_eq`), the correspondence Φ at each bond, linear, unital and multiplicative, pairing every insertion on the first chain with its bond operator on the second against all closed-chain words, with uniqueness (`MPSChainTensor.exists_insertionHom`, `MPSChainTensor.insertionHom_unique`, blueprint `lem:chain_exists_insertionHom`), and by Skolem–Noether the per-bond conjugation of full-length arc products (`MPSChainTensor.exists_conjugation_of_sameState`, blueprint `lem:chain_exists_conjugation_of_sameState`).

The capstone is the site-dependent closed-chain corollary at the source's bound: two window-injective site-dependent chains on $n \geq 2L+1$ sites generating the same state are gauge equivalent, one invertible gauge per bond, concluding the existing cyclic gauge-equivalence predicate (`TNLean.PEPS.fundamentalTheorem_normalMPSChain_of_overlap` with `MPSChainTensor.GaugeEquiv`, blueprint `cor:fundamentalTheorem_normalMPSChain_of_overlap`). The source displays the corollary only for translation-invariant tensors; the site-dependent assembly needs two steps the translation-invariant collapse does not: a window covariance derived from the conjugations at the two ends of an arc — the connecting matrix produced by the two-ends extraction is pinned to a nonzero scalar multiple of the far gauge through the centralizer of the matrix algebra — and a scalar absorption, where one circuit of the chain forces the product of the letterwise scalars to one so that the partial products dress the gauges consistently across the seam. These two steps are new mathematics relative to the translation-invariant route, not reindexing; everything else in the site-dependent development is the source's argument with arcs in place of word products.

One scope restriction remains, marked in the sources: all sites share one physical dimension d and all bonds one bond dimension D, the dimensions of the chain vocabulary of `TNLean/MPS/Chain/Defs.lean`, while the source poses no restriction on the local dimensions of the site-dependent tensors. The per-bond-dimension generality would replace the matrix algebra by rectangular matrix spaces with dimensions varying around the chain, and with it every spanning, trace-pairing and Skolem–Noether ingredient of the route; it remains a follow-up. With this, of the two items recorded as open at the end of the previous section, the site-dependent form of Lemma 5 is formalized up to the uniform-dimension restriction, and the $2L+1$ strengthening for two-dimensional tensor networks (the corollary at lines 2297–2318) remains open.

The $2L+1$ strengthening for two-dimensional tensor networks

The corollary at lines 2297–2318 — the torus theorem with one rectangle shape $L \times K$ assumed injective and the sizes $n \geq 2L+1$, $m \geq 2K+1$ — is the last open item of the section, and the source gives only a proof sketch for it. The sketch is now scoped in the dedicated note `docs/paper-gaps/peps_normal_ft_2d_overlap.tex`: the filled-in derivation runs a chain of $L + K$ overlapping $L \times K$ windows around each edge (sliding across the edge, then turning the corner), compares consecutive windows by inverting the complement of their union — the only step where the sizes enter — and ends on the diagonally offset pair of end windows, whose only joining edge is the reference edge itself, so the extracted bond operator is single-edge by the same single-crossing geometry the landed witness engine consumes. No research-level gap hides in the sketch; in particular no row-coherence question arises, because the argument never runs the one-dimensional theorem row by row. The landed three-block witness production cannot be reused at the corollary's sizes: at 5×5 with $L = K = 2$ the host block of the natural single-crossing pair contains a vertex lying in no 2×2 square inside the host, so host injectivity is not derivable, and the window chain is unavoidable. The note records the layer plan; the geometry layer is the first installment, and the corollary itself remains unformalized.

The investigation has since reached its definitive endpoint, which supersedes the optimistic reading above ("no research-level gap hides in the sketch"). The dedicated note now carries a consolidated account. Its findings: the geometry and state-equality layers are fully formalized at the corollary's minimal size, ending in the open-boundary end-pair equality `TNLean.PEPS.staircasePair_insert_eq_open`, which never inverts the non-injective complement of the end pair. The development is gated on a single input, the forward-transfer multiplicativity at the reference edge — the cross-bond pushing of one virtual operation across the window chain. That input is the one step the source's sketch asserts by analogy to Lemma 5 rather than derives, and the analogy fails dimensionally: Lemma 5's closing read-off relies on a one-dimensional window product being a square matrix on its bond, whereas a two-dimensional $L \times K$ window touches the reference edge with one endpoint and gives a vector family, never a square-matrix span. Every route to the missing input is machine-checked impossible at the minimal size — single-vertex injectivity (normality is weaker, not stronger), the one-dimensional port (non-square window), the end-pair coarse frame (third region not injective), the per-bond coarse frame (`TNLean.PEPS.horizontalStaircasePair_ne_arcRectangle`), and the row-and-column reduction (`TNLean.PEPS.RowColumnReductionObstruction.rowCutGaugeFactorizes_false`); the corollary's translation invariance removes only the interior-bond rescaling sub-obstruction. The corollary therefore remains unformalized and unclaimed, closeable only by a new idea evading these impossibilities or by the source supplying the cross-bond-pushing argument its sketch only asserts.

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.