From the Gibbs Weight to the Spectral Gap: A Machine-Checked Transfer-Matrix Identification and Spectral-Gap Chain for the Open $\mathbb{Z}_2$ Ising Spin Chain

Verbatim Lean 4 definitions, exact machine-checked vs. analytic boundaries, and reproducible axiom-oracle audit

Lluis Eriksson

23 November 2026 (v3)

Abstract

We report a Lean 4 / `mathlib` formalization of the transfer-matrix identification and spectral-gap chain for the one-dimensional open-boundary $\mathbb{Z}_2$ Ising spin chain (one $\mathbb{Z}_2$ spin per time slice, 2×2 transfer operator): from the Boltzmann path weight $W_\beta(\sigma) = \prod_{t=0}^{n-1} e^{\beta s(\sigma_t,\sigma_{t+1})}$ and an independently defined normalized transfer operator T_β on $\mathbb{C}^2$, through the computed partition function $Z_n = 2(e^\beta + e^{-\beta})^n$ and the Hermitian identification theorem $\mathbb{E}_n[\mathrm{conj}(A(\sigma_0))\, B(\sigma_n)] = \langle A\Omega,\, T_\beta^n(B\Omega)\rangle_{\mathbb{C}}$ (using Mathlib's inner product, conjugate-linear in the first slot), to the exact two-point correlation $\mathbb{E}_n[f(\sigma_0)f(\sigma_n)] = (\tanh\beta)^n$ of the real-valued sign observable f.

We separate explicitly between *machine-checked Lean 4 declarations* (quoted verbatim from `Z2Transfer.lean`, `SharpBridge.lean`, and `Z2Identification.lean`, together with their presence in `oracle_check.lean`) and *elementary analytic consequences proved in prose*: specifically, Lean proves the operator norm bound $\|T_\beta - |\Omega\rangle\langle\Omega|\| \le \tanh\beta$ for $\beta \ge 0$ and the scalar positivity inequalities $0 < 1 - \tanh\beta$ $(\forall\beta)$ and $0 < -\log(\tanh\beta)$ $(\beta > 0)$, while the exact all-β eigenvalue spectrum $\{1, \tanh\beta\}$, the distinction at $\beta < 0$ between the *absolute spectral gap* $1 - |\tanh\beta|$ and the *algebraic eigenvalue gap* $1 - \tanh\beta$, and the rank-one nonexistence of a finite Hermitian H with $T_0 = e^{-H}$ at $\beta = 0$ are elementary analytic consequences established in Section 3. We further clarify that `z2_pairing_nondegenerate` covers only time-zero single-slice observables, whereas the full positive-time Osterwalder–Seiler GNS quotient is nontrivial (witnessed by $F(\sigma_0,\sigma_1) = f(\sigma_1) - (\tanh\beta)f(\sigma_0)$) and is formalized separately (`ARR-2026-2TBRDN2DE598KSXD`), and that companion reflection positivity (`ARR-2026-10JJK0VCCS87VTQH`) is not a logical dependency of this module. Nothing here is a claim about SU(N), spatial-volume uniformity, continuum limits, or the Clay Yang–Mills problem.

1 Introduction and exact scope

Constructive lattice field theory specifies a Euclidean probability measure, whereas a mass gap is a spectral property of a self-adjoint transfer operator. Connecting the two requires an *identification theorem* expressing Euclidean expectations in the measure as matrix elements of powers of the transfer operator, proved without circularity (without defining the measure from the operator).

For the open-boundary 1D $\mathbb{Z}_2$ Ising spin chain on $n+1$ time slices $\sigma : \{0, \ldots, n\} \to \{0, 1\}$, we define solely from exp and the bond sign $s(i, j) = +1$ $(i = j)$, -1 $(i \neq j)$:

$$W_\beta(\sigma) = \prod_{t=0}^{n-1} e^{\beta s(\sigma_t, \sigma_{t+1})}, \qquad Z_n(\beta) = \sum_\sigma W_\beta(\sigma),$$
$$\mathbb{E}_n\big[\mathrm{conj}(A(\sigma_0))\, B(\sigma_n)\big] = \frac{1}{Z_n(\beta)} \sum_\sigma \mathrm{conj}(A(\sigma_0))\, B(\sigma_n)\, W_\beta(\sigma).$$

(We write conj$(\cdot)$ explicitly rather than an overline so plain-text PDF extraction never drops complex conjugation.) Independently, on $\mathbb{C}^2$ with Mathlib's Euclidean inner product $\langle x, y\rangle_{\mathbb{C}} = \sum_{i\in\{0,1\}} \mathrm{conj}(x_i)\, y_i$ (conjugate-linear in the first slot, linear in the second), we define

$$T_\beta = a(\beta)\, \mathbf{1} + b(\beta)\, S, \qquad a(\beta) = \frac{e^\beta}{e^\beta + e^{-\beta}}, \quad b(\beta) = \frac{e^{-\beta}}{e^\beta + e^{-\beta}},$$

where S swaps the two coordinates.

Scope boundaries.

1. **Open 1D $\mathbb{Z}_2$ Ising spin chain, not 1D pure-link gauge theory.** An open one-dimensional pure-link gauge theory has no plaquettes and every link configuration is gauge-equivalent; the model formalized here is the open 1D $\mathbb{Z}_2$ Ising spin chain.

2. **Transfer-matrix identification, not full positive-time OS reconstruction.** The companion reflection-positivity module (`ARR-2026-10JJK0VCCS87VTQH`) is *not* imported or consumed by `Z2Identification.lean`.

3. **Temporal vs. spatial uniformity.** The exact identity $(\tanh\beta)^n$ holds for every temporal separation $n \in \mathbb{N}$; spatial-volume uniformity does not arise because each spatial slice is a single site $(\dim \mathbb{C}^2 = 2)$.

4. **Classical mathematics.** The 1D Ising transfer matrix is textbook [2]; the candidate contribution is a replayable, non-circular Lean 4 formalization connecting the Gibbs path sum to operator clustering. No exhaustive priority search over all external Lean/Isabelle/Coq Ising formalizations is claimed, and nothing here bears on SU(N), continuum limits, or the Clay problem [3].

2 Verbatim Lean 4 definitions and checked theorem signatures

To make every definition and every machine-checked statement auditable directly from this PDF, we quote verbatim the complete definitions and theorem signatures from `YangMills/OS/SharpBridge.lean`, `YangMills/OS/Z2Transfer.lean`, and `YangMills/OS/Z2Identification.lean` at archival release `yang-mills-archive-v1.0` (commit `6b54eca695e96b3038ceda18276c4e28fb45983f`, where these files are byte-identical to their original freeze checkpoint `bd2c18394ce8a55ec73413c11a9f311bbf002248`):

```
-- YangMills/OS/SharpBridge.lean (lines 67-82)
noncomputable def vacuumProjectionC (Omega : H) : H ->L[Complex] H :=
  (innerSL Complex Omega).smulRight Omega
noncomputable def projectedTransferC (T : H ->L[Complex] H) (Omega : H) : H ->L[Complex] H :=
```

```
  T - vacuumProjectionC Omega
noncomputable def connCorrC (T : H →L[Complex] H) (Omega : H) (v : H) (n : Nat) : Complex :=
  inner Complex v ((T ^ n) v) - inner Complex v Omega * inner Complex Omega v

-- YangMills/OS/Z2Transfer.lean (lines 74-165, 261-355)
noncomputable def invSqrtTwo : Real := (Real.sqrt 2)^(-1)
noncomputable def swapLin : EuclideanSpace Complex (Fin 2) →l[Complex]
    EuclideanSpace Complex (Fin 2) where
  toFun := fun v => WithLp.toLp 2 (fun i => v.ofLp (1 - i)) ...
noncomputable def swapOp : EuclideanSpace Complex (Fin 2) →L[Complex]
    EuclideanSpace Complex (Fin 2) := LinearMap.toContinuousLinearMap swapLin
noncomputable def z2Norm (beta : Real) : Real := Real.exp beta + Real.exp (-beta)
noncomputable def z2A (beta : Real) : Real := Real.exp beta / z2Norm beta
noncomputable def z2B (beta : Real) : Real := Real.exp (-beta) / z2Norm beta
noncomputable def z2TransferOp (beta : Real) :
    EuclideanSpace Complex (Fin 2) →L[Complex] EuclideanSpace Complex (Fin 2) :=
  (z2A beta : Complex) • 1 + (z2B beta : Complex) • swapOp
noncomputable def z2Vac : EuclideanSpace Complex (Fin 2) :=
  WithLp.toLp 2 (fun _ => ((invSqrtTwo : Real) : Complex))

theorem z2TransferOp_fix (beta : Real) : z2TransferOp beta z2Vac = z2Vac
theorem z2TransferOp_selfAdjoint (beta : Real) : IsSelfAdjoint (z2TransferOp beta)
theorem z2Projected_eq (beta : Real) (v : EuclideanSpace Complex (Fin 2)) (i : Fin 2) :
    (projectedTransferC (z2TransferOp beta) z2Vac v).ofLp i
      = ((z2A beta - z2B beta : Real) : Complex) * (v.ofLp i - (v.ofLp 0 + v.ofLp 1) / 2)
theorem z2TransferOp_gap {beta : Real} (hbeta : 0 <= beta) :
    ||projectedTransferC (z2TransferOp beta) z2Vac|| <= z2A beta - z2B beta
theorem z2_clustering_of_transfer {beta : Real} (hbeta : 0 <= beta)
    (v : EuclideanSpace Complex (Fin 2)) :
    Exists fun C : Real => forall n : Nat,
      ||connCorrC (z2TransferOp beta) z2Vac v n|| <= C * (z2A beta - z2B beta) ^ n
theorem z2_spectral_gap_pos (beta : Real) : 0 < 1 - Real.tanh beta
theorem z2_mass_pos {beta : Real} (hbeta : 0 < beta) : 0 < -Real.log (Real.tanh beta)

-- YangMills/OS/Z2Identification.lean (lines 67-112, 221-422)
def z2Sign (i j : Fin 2) : Real := if i = j then 1 else -1
noncomputable def z2Bond (beta : Real) (i j : Fin 2) : Real :=
  Real.exp (beta * z2Sign i j)
noncomputable def z2PathWeight (beta : Real) {n : Nat}
    (sigma : Fin (n + 1) -> Fin 2) : Real :=
  Finset.prod Finset.univ fun t : Fin n =>
    z2Bond beta (sigma t.castSucc) (sigma t.succ)
noncomputable def z2Partition (beta : Real) (n : Nat) : Real :=
  Finset.sum Finset.univ fun sigma : Fin (n + 1) -> Fin 2 => z2PathWeight beta sigma
noncomputable def z2PathSum (beta : Real) (n : Nat) (A B : Fin 2 -> Complex) : Complex :=
  Finset.sum Finset.univ fun sigma : Fin (n + 1) -> Fin 2 =>
    (starRingEnd Complex) (A (sigma 0)) * B (sigma (Fin.last n))
      * ((z2PathWeight beta sigma : Real) : Complex)
noncomputable def z2Expect (beta : Real) (n : Nat) (A B : Fin 2 -> Complex) : Complex :=
  z2PathSum beta n A B / ((z2Partition beta n : Real) : Complex)
noncomputable def z2Vec (A : Fin 2 -> Complex) : EuclideanSpace Complex (Fin 2) :=
  WithLp.toLp 2 A
noncomputable def z2Obs (A : Fin 2 -> Complex) : EuclideanSpace Complex (Fin 2) :=
  ((invSqrtTwo : Real) : Complex) • z2Vec A
def z2SignObs : Fin 2 -> Complex := fun i => if i = 0 then 1 else -1

theorem z2Partition_eq (beta : Real) (n : Nat) :
    z2Partition beta n = 2 * (z2Norm beta) ^ n
```

```
theorem z2_identification (beta : Real) (n : Nat) (A B : Fin 2 -> Complex) :
    z2Expect beta n A B = inner Complex (z2Obs A) (((z2TransferOp beta) ^ n) (z2Obs B))
theorem z2Vec_signObs_eigen (beta : Real) :
    z2TransferOp beta (z2Vec z2SignObs) = ((z2A beta - z2B beta : Real) : Complex) • z2Vec z2SignObs
theorem z2_two_point_tanh (beta : Real) (n : Nat) :
    z2Expect beta n z2SignObs z2SignObs = ((Real.tanh beta : Real) : Complex) ^ n
theorem z2_one_point_sign_eq_zero (beta : Real) (n : Nat) :
    z2Expect beta n z2SignObs (fun _ => (1 : Complex)) = 0
theorem z2_pairing_nondegenerate (beta : Real) {A : Fin 2 -> Complex}
    (h : z2Expect beta 0 A A = 0) : A = fun _ => 0
```

Remark 2.1 (Non-circularity check). The measure-side dependency closure is `z2Expect` $\to$ $\{$`z2PathSum`, `z2Partition`$\}$ $\to$ `z2PathWeight` $\to$ `z2Bond` $\to$ $\{\exp,$ `z2Sign`$\}$, containing no reference to `z2TransferOp`, `z2A`, or `z2B`. The two sides meet at a single bridge lemma `z2Bond_eq` expressing $k_\beta(i,j) = (e^\beta + e^{-\beta})(a\delta_{ij} + b(1-\delta_{ij}))$.

3 Machine-checked spectral theorems vs. analytic consequences

To avoid any ambiguity between kernel-checked declarations and standard mathematical deductions in prose, we separate the two explicitly.

Theorem 3.1 (Machine-checked spectral statements in `Z2Transfer.lean` and `Z2Identification.lean`)**.** *The Lean 4 kernel verifies (with zero* `sorry` *and standard axioms only):*

1. $T_\beta\Omega = \Omega$, $T_\beta^* = T_\beta$, *and* $T_\beta f = (\tanh\beta)f$ *for all* $\beta \in \mathbb{R}$ *(*`z2TransferOp_fix`*,* `z2TransferOp_selfAdjoint`*,* `z2Vec_signObs_eigen`*,* `z2A_sub_z2B_eq_tanh`*);*
2. *the exact componentwise identity* $(T_\beta - |\Omega\rangle\langle\Omega|)v = (\tanh\beta)(v - \langle\Omega, v\rangle_{\mathbb{C}}\Omega)$ *for all* $\beta \in \mathbb{R}$ *(*`z2Projected_eq`*);*
3. *the one-sided operator-norm bound* $\|T_\beta - |\Omega\rangle\langle\Omega|\| \le \tanh\beta < 1$ *for* $\beta \ge 0$ *(*`z2TransferOp_gap`*,* `z2A_sub_z2B_lt_one`*);*
4. *the scalar inequalities* $0 < 1 - \tanh\beta$ *for all* $\beta \in \mathbb{R}$ *(*`z2_spectral_gap_pos`*) and* $0 < -\log(\tanh\beta)$ *for* $\beta > 0$ *(*`z2_mass_pos`*);*
5. *the clustering corollary* `z2_clustering_of_transfer` *for* $\beta \ge 0$.

Proposition 3.2 (Elementary analytic consequences across $\beta \in \mathbb{R}$, proved in prose)**.** *The following standard spectral statements follow immediately from the* 2×2 *eigendecomposition* $T_\beta = P + (\tanh\beta)(\mathbf{1} - P)$ *(where* $P = |\Omega\rangle\langle\Omega|$, $\mathbf{1} - P = |e_-\rangle\langle e_-|$ *with* $e_- = 2^{-1/2}(1,-1)$*), and are* not *separately formalized as operator-logarithm theorems in Lean:*

1. ***Exact spectrum:*** $\mathrm{spec}(T_\beta) = \{1, \tanh\beta\}$ *for every* $\beta \in \mathbb{R}$*, and* $\|T_\beta - P\| = |\tanh\beta|$ *for every* $\beta \in \mathbb{R}$.
2. ***Positive coupling (*$\beta > 0$*):*** $0 < \tanh\beta < 1$*, so* T_β *is strictly positive definite and admits the unique Hermitian logarithm* $H = -\log(T_\beta) = (-\log\tanh\beta)(\mathbf{1} - P)$ *satisfying* $T_\beta = e^{-H}$*, with Hamiltonian mass gap* $m(\beta) = -\log(\tanh\beta) > 0$.

3. ***Negative coupling*** *($\beta < 0$):* *the non-vacuum eigenvalue* $\tanh\beta$ *is negative, so* T_β *has no real self-adjoint logarithm* H *with* $T_\beta = e^{-H}$. *Here one must distinguish the* absolute spectral gap $1 - |\tanh\beta| = 1 + \tanh\beta \in (0,1)$ *(which governs the contraction rate* $\|T_\beta - P\| = |\tanh\beta|$*) from the* algebraic eigenvalue gap $1 - \tanh\beta \in (1,2)$ *(for instance, at* $\tanh\beta = -1/2$*, the absolute spectral gap is* $1/2$ *while the algebraic eigenvalue gap is* $3/2$*). The Lean lemma* `z2_spectral_gap_pos` *states only the scalar inequality* $0 < 1 - \tanh\beta$.

4. ***Zero coupling*** *($\beta = 0$):* $\tanh 0 = 0$, *so* $T_0 = P$ *is rank-one and singular; no finite Hermitian operator* H *on* $\mathbb{C}^2$ *satisfies* $e^{-H} = T_0$ *(as* e^{-H} *is always invertible). For every* $n \geq 1$, *the sign two-point correlation* $\mathbb{E}_n[f(\sigma_0)f(\sigma_n)] = 0$ *and all connected two-point correlations across distinct sites vanish identically (whereas uncentered expectations of constant observables remain* $\mathbb{E}_n[1 \cdot 1] = 1$*). Restricting* `z2_mass_pos` *to* $\beta > 0$ *avoids Lean's junk value* `Real.log 0 = 0`.

4 The identification theorem and exact correlation

Theorem 4.1 (Hermitian transfer-matrix identification; `z2_identification`)**.** *For every* $\beta \in \mathbb{R}$, *every* $n \in \mathbb{N}$, *and all* $A, B : \{0,1\} \to \mathbb{C}$,

$$\mathbb{E}_n\big[\mathrm{conj}(A(\sigma_0))\, B(\sigma_n)\big] \;=\; \langle A\Omega,\; T_\beta^n(B\Omega)\rangle_{\mathbb{C}},$$

where $\langle x, y\rangle_{\mathbb{C}} = \sum_{i\in\{0,1\}} \mathrm{conj}(x_i)\, y_i$ *is Mathlib's Euclidean inner product, conjugate-linear in the first argument (`z2_identification`).*

Proof, as formalized. Split the sum over $\sigma : \mathrm{Fin}(n+1) \to \mathrm{Fin}\,2$ at the first spin $\sigma = (s,\tau)$. Factoring out the first bond $k_\beta(s,\tau_0) = (e^\beta + e^{-\beta})(T_\beta)_{s,\tau_0}$ and using self-adjointness of T_β yields by induction on n (`z2PathSum_succ`):

$$\sum_\sigma \mathrm{conj}(A(\sigma_0))\, B(\sigma_n)\, W_\beta(\sigma) = (e^\beta + e^{-\beta})^n\, \langle A,\, T_\beta^n B\rangle_{\mathbb{C}}.$$

Setting $A = B \equiv 1$ and using $T_\beta \mathbf{1} = \mathbf{1}$ gives $Z_n(\beta) = 2(e^\beta + e^{-\beta})^n$ (`z2Partition_eq`). Dividing cancels $(e^\beta + e^{-\beta})^n$ and leaves $\frac{1}{2}\langle A, T_\beta^n B\rangle_{\mathbb{C}} = \langle 2^{-1/2}A,\, T_\beta^n(2^{-1/2}B)\rangle_{\mathbb{C}} = \langle A\Omega,\, T_\beta^n(B\Omega)\rangle_{\mathbb{C}}$. (Without $\mathrm{conj}(\cdot)$ on $A(\sigma_0)$, the identity fails already at $n = 0$ for $A \equiv i$, $B \equiv 1$, where $\mathbb{E}_0[i \cdot 1] = i \neq -i = \langle i\Omega, \Omega\rangle_{\mathbb{C}}$.) □

Corollary 4.2 (Exact sign two-point function; `z2_two_point_tanh`)**.** *For the real-valued sign observable* $f(0) = +1$, $f(1) = -1$ *(so* $\mathrm{conj}(f(\sigma)) = f(\sigma)$*), every* $\beta \in \mathbb{R}$, *and every* $n \in \mathbb{N}$,

$$\mathbb{E}_n[f(\sigma_0)f(\sigma_n)] = (\tanh\beta)^n, \qquad \mathbb{E}_n[f(\sigma_0)] = 0$$

(`z2_two_point_tanh`, `z2_one_point_sign_eq_zero`).

Remark 4.3 (Portability to general finite kernels). For a general finite symmetric kernel K, path induction yields $Z_n = \mathbf{1}^\top K^n \mathbf{1}$. The power formula $Z_n = |S|\,\lambda^n$ and the uniform vacuum $\Omega = |S|^{-1/2}\mathbf{1}$ require *constant row sums* ($K\mathbf{1} = \lambda\mathbf{1}$), not symmetry alone.

5 Time-zero definiteness, the positive-time OS quotient, and clustering

Remark 5.1 (Time-zero definiteness vs. the full positive-time OS quotient). The $n = 0$ instance of Theorem 4.1 gives $\mathbb{E}_0[\mathrm{conj}(A(\sigma_0))A(\sigma_0)] = \|A\Omega\|^2$, so `z2_pairing_nondegenerate` proves that the pairing is strictly positive definite on *time-zero single-slice observables* $A(\sigma_0)$. This does *not* mean that the Osterwalder–Seiler reflection pairing on the full positive-time algebra $\{\sigma_0, \sigma_1, \dots\}$ is definite: under site reflection across $t = 0$, the nonzero observable $F(\sigma_0, \sigma_1) := f(\sigma_1) - (\tanh\beta) f(\sigma_0)$ satisfies $\mathbb{E}[F \mid \sigma_0] = 0$ for every finite β, hence has zero reflected norm $\mathbb{E}[\mathrm{conj}(\theta F)\, F] = 0$ by Markov conditional independence across σ_0. Thus the full positive-time GNS quotient is nontrivial even for the 1D Ising chain and is formalized separately (`ARR-2026-2TBRDN2DE598KSXD`).

Remark 5.2 (Algebraic structure of T_β and role of the abstract clustering criterion). With $P = |\Omega\rangle\langle\Omega|$ and $r = \tanh\beta = a - b$, algebraically $T_\beta = P + r(\mathbf{1} - P) = r\mathbf{1} + (1 - r)P$. The distinction from earlier toy witnesses is *constructive*, not algebraic: T_β is built from the Boltzmann bond weights $e^{\pm\beta}$ rather than taking a target rate r as input. Finally, `z2_two_point_tanh` is proved directly from the eigenfunction $T_\beta f = (\tanh\beta) f$; the separate corollary `z2_clustering_of_transfer` feeds T_β into the abstract dense-family clustering criterion `connCorrC_le_of_gap` (`ARR-2026-31ED4MTKC18JPSDE`) for arbitrary $v \in \mathbb{C}^2$ and $\beta \geq 0$, rather than serving as a dependency of Corollary 4.2.

6 Source-pinned declaration manifest and reproducibility

All Lean files live in https://github.com/lluiseriksson/THE-ERIKSSON-PROGRAMME. Both `YangMills/OS/Z2Transfer.lean` and `YangMills/OS/Z2Identification.lean` were frozen at commit `bd2c18394ce8a55ec73413c11a9f311bbf002248` and remain byte-identical at archival release `yang-mills-archive-v1.0` (commit `6b54eca695e96b3038ceda18276c4e28fb45983f`, Lean `v4.29.0-rc6`, pinned Mathlib `07642720480157414db592fa85b626dafb71355b`). Every theorem in Theorem 3.1 and Section 4 compiles inside `YangMillsCore`, and the five headline theorems (`z2Partition_eq`, `z2_identification`, `z2_two_point_tanh`, `z2TransferOp_gap`, `z2_mass_pos`) are audited directly by `#print axioms` inside the root `oracle_check.lean` validated by `scripts/check_oracle_output.py`.

Reproduce the build and verify the axiom output locally (on Windows include `-c core.longpaths=true`):

```
git clone -c core.longpaths=true \
  https://github.com/lluiseriksson/THE-ERIKSSON-PROGRAMME.git
cd THE-ERIKSSON-PROGRAMME && git checkout yang-mills-archive-v1.0
lake exe cache get && lake build YangMillsCore
lake env lean oracle_check.lean > oracle-output.txt
python scripts/check_oracle_output.py --output oracle-output.txt
```

GitHub Actions (`lean-build.yml`) runs this exact build-and-oracle check on `main`, rejecting any `sorryAx` or non-standard axiom outside `[propext, Classical.choice, Quot.sound]`.

Disclosure

Large language models assisted Lean 4 formalization and manuscript editing under author supervision; every theorem in Section 2 is checked by the Lean 4 kernel. Author Lluis Eriksson

administers AIRR (https://airr.science/); no independent editorial admission or human peer review is claimed.

References

[1] K. Osterwalder and E. Seiler, *Gauge field theories on a lattice*, Annals of Physics **110** (1978), 440–471.

[2] B. Simon, *The Statistical Mechanics of Lattice Gases, Volume I*, Princeton University Press, 1993.

[3] A. Jaffe and E. Witten, *Quantum Yang–Mills theory*, Clay Mathematics Institute Millennium Prize Problem description, 2000.

[4] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, CADE-28, LNCS 12699 (2021), 625–635.