Recoverability length scales and Wilson loops in lattice gauge theories

Protocol, definitions, and conjectural links to confinement diagnostics

Lluis Eriksson
Independent Researcher
lluiseriksson@gmail.com

January 2026

Abstract

We propose a numerical protocol and a set of falsifiable conjectures relating quantum-information recoverability measures to confinement diagnostics in lattice gauge theories. For a tripartition A–B–C on a finite lattice and a collar width w, we define a Petz-type recovery error $E_{\rm rec}^{\rm Petz}(w)$ based on fidelity and extract an associated recoverability length scale from threshold and fit-based criteria. Since gauge constraints obstruct naive subsystem factorization, we formulate the protocol in an extended-Hilbert-space (EHS) prescription by default, while also outlining an algebraic (gauge-invariant observable) variant and its subtleties (centers, sector decompositions). We specify a practical regularization for Petz-type reconstruction when ρ_B is not full rank and discuss normalization conventions in numerics. We then formulate conjectures that $E_{\rm rec}^{\rm Petz}(w)$ decays exponentially in gapped phases and that its decay scale tracks confinement scales set by Wilson loops, such as an effective string tension extracted from area-law fits or Creutz ratios. We do not claim a theorem-level confinement result; the paper is a self-contained protocol and roadmap intended to be tested on small lattices (exact diagonalization/tensor networks) and, where reduced states are accessible, by Monte Carlo plus replica methods. As a control experiment we include TFIM data validating the pipeline and illustrating the growth of $\xi_{\rm rec}$ near criticality.

Contents

1 Introduction

Our companion note *Petz recoverability in AQFT via conditional expectations: a framework and a conditional exponential recovery bound* [1] isolated an AQFT-correct Petz-recoverability framework in type III settings and made explicit which "bridge assumptions" would imply exponential recoverability in the continuum. Here we focus on a finite-dimensional arena: lattice gauge theories, where recovery maps and reduced states can be implemented explicitly (with due care for gauge constraints).

Goal. Define and measure a *recoverability length* on the lattice and relate it to standard gauge-theory diagnostics, in particular the behavior of Wilson loops.

Scope and non-claims. We do not prove confinement, nor the Yang–Mills mass gap. We provide definitions, a reproducible protocol, and conjectures designed to be falsifiable by numerical data.

2 Finite-dimensional preliminaries

2.1 Fidelity and recovery error

For density matrices ρ and σ on a finite-dimensional Hilbert space, we use the squared Uhlmann fidelity

$$F(\rho,\sigma) := \left(\mathrm{Tr}\sqrt{\sqrt{\rho}\,\sigma\,\sqrt{\rho}}\right)^2 \in [0,1],$$

and define the recovery error

$$E_{\mathrm{rec}}(\rho \to \sigma) := -\log F(\rho,\sigma).$$

2.2 Tripartitions and collar width

Let $\mathcal{L}$ be a finite lattice. Fix a tripartition of degrees of freedom into disjoint sets A, B, C, where B is a buffer (collar) separating A from C. We measure collar width w as graph distance between the boundaries of A and C (in units of links, unless otherwise stated). Unless explicitly stated otherwise, all lengths and areas are measured in lattice units at fixed lattice spacing.

3 Subsystem prescriptions in gauge theory

Gauge constraints obstruct naive factorization of the physical Hilbert space across spatial regions. We outline two prescriptions.

3.1 Extended Hilbert space (EHS) prescription (default)

Definition 3.1 (EHS prescription)**.** Let $\mathcal{H}_{\mathrm{phys}}$ be the gauge-invariant Hilbert space (Hamiltonian formulation). An EHS prescription specifies:

1. an embedding $\iota : \mathcal{H}_{\mathrm{phys}} \hookrightarrow \mathcal{H}_{\mathrm{ext}}$ into an extended tensor-product Hilbert space $\mathcal{H}_{\mathrm{ext}} \cong \mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$ (with edge modes on cut links),
2. a rule to compute reduced density matrices by partial trace in $\mathcal{H}_{\mathrm{ext}}$.

For a physical density matrix ρ_{phys} we define $\rho^{\mathrm{EHS}}_{ABC} := \iota\,\rho_{\mathrm{phys}}\,\iota^\dagger$ (which has unit trace if ι is an isometry; otherwise we renormalize by its trace) and its marginals ρ^{EHS}_{AB}, ρ^{EHS}_{B}, ρ^{EHS}_{BC} by partial trace.

Remark 3.2 (Operational choice)*.* All default definitions and the numerical protocol in this paper are formulated in the EHS prescription, so that all reduced objects are ordinary density matrices and F is the standard Uhlmann fidelity. The algebraic prescription below is included as a gauge-invariant alternative and for future refinements.

3.2 Algebraic (gauge-invariant observable) prescription (outline)

Definition 3.3 (Algebraic prescription (outline))**.** Let $\mathcal{A}(R)$ denote the gauge-invariant observable algebra associated with a region R (a finite-dimensional von Neumann algebra, typically with a nontrivial center). A global state defines reduced states by restriction to these subalgebras. A full algebraic implementation would define fidelity and recovery maps at the algebra level, possibly sector-wise across the center.

Remark 3.4 (Centers and sector decompositions)*.* In gauge theories, $\mathcal{A}(R)$ may have a center, corresponding to superselection sectors associated with boundary data. In such cases, "reduced states" are naturally block-diagonal across sectors, and fidelity/recovery may be implemented sector-wise and then combined. We do not develop the full algebraic variant here.

4 Petz recovery on the lattice

4.1 Regularized Petz-type reconstruction map

Let ρ_{ABC} be a density matrix on $\mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_C$ (in the chosen subsystem prescription, by default EHS). Write $\rho_B = \mathrm{Tr}_{AC}(\rho_{ABC})$ and $\rho_{BC} = \mathrm{Tr}_A(\rho_{ABC})$.

Definition 4.1 (Regularized Petz-type map)**.** Fix a regularization parameter $\delta > 0$ and define $\rho_{B,\delta} := \rho_B + \delta\,\mathbf{1}_B$. Define the regularized Petz-type map $\mathcal{R}^{\mathrm{Petz}(\delta)}_{B\to BC}$ by

$$\mathcal{R}^{\mathrm{Petz}(\delta)}_{B\to BC}(X_B) := \rho_{BC}^{1/2}\left(\rho_{B,\delta}^{-1/2}\,X_B\,\rho_{B,\delta}^{-1/2}\otimes \mathbf{1}_C\right)\rho_{BC}^{1/2}.$$

Remark 4.2 (Trace preservation and normalization)*.* The map $\mathcal{R}^{\mathrm{Petz}(\delta)}_{B\to BC}$ is completely positive. For $\delta > 0$ it is generally not exactly trace-preserving (unlike the ideal Petz map defined using $\rho_B^{-1/2}$ on the support of ρ_B). In numerical implementations we therefore interpret $\mathcal{R}^{\mathrm{Petz}(\delta)}$ as a stable Petz-type reconstruction map; when computing fidelities we normalize the reconstructed output state by its trace. Our reported results track the stability of extracted scales under δ variations.

Remark 4.3*.* Unless stated otherwise, we use a fixed global $\delta = 10^{-12}$ (double precision) for all collar widths w and verify stability under $\delta \in [10^{-14}, 10^{-10}]$. Alternatively, one may set $\delta = \delta(w)$ adaptively from a spectral cutoff on $\rho_B(w)$.

4.2 Petz recoverability profile and length scales

Definition 4.4 (Petz recoverability profile)**.** Fix a subsystem prescription (default: EHS) and a regularization rule for $\mathcal{R}^{\mathrm{Petz}}$ (Definition 4.1). For a family of tripartitions $(A, B(w), C)$ indexed by collar width w, define the (possibly unnormalized) reconstructed operator

$$\widetilde{\rho}_{ABC}(w) := (\mathrm{id}_A \otimes \mathcal{R}^{\mathrm{Petz}(\delta)}_{B\to BC})(\rho_{AB}),$$

and set $\widehat{\rho}_{ABC}(w) := \widetilde{\rho}_{ABC}(w)/\mathrm{Tr}(\widetilde{\rho}_{ABC}(w))$ whenever $\mathrm{Tr}(\widetilde{\rho}_{ABC}(w)) > 0$. Define

$$E^{\mathrm{Petz}}_{\mathrm{rec}}(w) := -\log F\Big(\rho_{ABC},\, \widehat{\rho}_{ABC}(w)\Big).$$

All reduced states ρ_{ABC}, ρ_{AB}, ρ_B, ρ_{BC} are computed in the same subsystem prescription.

Definition 4.5 (Threshold recoverability length)**.** Fix $\varepsilon > 0$. Define

$$d_{\mathrm{rec}}(\varepsilon) := \min\{\, w : \; E_{\mathrm{rec}}^{\mathrm{Petz}}(w) \le \varepsilon \,\},$$

whenever the set is nonempty.

Definition 4.6 (Recoverability rate)**.** Assuming $E_{\mathrm{rec}}^{\mathrm{Petz}}(w)$ is approximately exponential on a fitting window $w \in [w_{\min}, w_{\max}]$, define ξ_{rec} by the slope of a linear fit

$$\log E_{\mathrm{rec}}^{\mathrm{Petz}}(w) \approx \log A - \frac{w}{\xi_{\mathrm{rec}}}.$$

5 Wilson loops and confinement diagnostics

5.1 Wilson loops and effective string tension

Given a closed loop ℓ on the lattice, let $W(\ell)$ denote the Wilson-loop operator and $\langle W(\ell)\rangle$ its expectation in the state of interest.

In a confining regime one expects, for large loops,

$$\langle W(\ell)\rangle \approx \exp\bigl(-\sigma \,\mathrm{Area}(\ell) - \mu \,\mathrm{Perimeter}(\ell)\bigr),$$

where σ is the string tension (in lattice units at fixed lattice spacing).

Remark 5.1. Here $\mathrm{Area}(\ell)$ is the number of plaquettes in a chosen minimal surface spanning ℓ (in lattice units), and $\mathrm{Perimeter}(\ell)$ is the loop length measured in links.

5.2 Creutz ratios (practical proxy)

On finite lattices, a common proxy for σ is a Creutz ratio built from rectangular loops, yielding an effective string tension σ_{eff}.

6 Concrete minimal testbed

6.1 $\mathbb{Z}_2$ lattice gauge theory (Hamiltonian)

As a minimal reproducible testbed, consider a $\mathbb{Z}_2$ lattice gauge theory in the Hamiltonian formulation on a small two-dimensional spatial lattice (i.e. $2+1$ spacetime dimensions), with qubits on links. A standard Kogut–Susskind Hamiltonian [4] is

$$H(g) = -g \sum_{\ell} \sigma_\ell^x \; - \; \frac{1}{g} \sum_{p} \prod_{\ell \in \partial p} \sigma_\ell^z,$$

where the first sum runs over links ℓ and the second over plaquettes p.

6.2 Tripartition on links

Choose A and C as sets of links separated by a buffer set $B(w)$ (also links). In the EHS prescription, cut links across the boundaries and introduce edge modes. Then compute $E_{\mathrm{rec}}^{\mathrm{Petz}}(w)$ and Wilson-loop proxies on the same finite lattice.

7 Conjectures and falsifiable predictions

Remark 7.1 (Asymptotic notation). We write $f(w) \lesssim g(w)$ if there exists $C > 0$ such that $f(w) \leq C\, g(w)$ in the fitting range. We write $x \asymp y$ if $c_1 y \leq x \leq c_2 y$ for constants $c_1, c_2 > 0$ (in the regime under discussion).

Conjecture 7.2 (Exponential Petz recoverability in gapped phases)**.** In a gapped phase (finite correlation length), for fixed region sizes of A and C and varying collar width w, the Petz recoverability profile obeys

$$E_{\mathrm{rec}}^{\mathrm{Petz}}(w) \lesssim A\, e^{-w/\xi_{\mathrm{rec}}}$$

on a suitable window of w.

Conjecture 7.3 (Recoverability tracks confinement scales)**.** In a confining phase, the recoverability length ξ_{rec} (Definition 4.6) tracks a confinement scale set by the string tension extracted from Wilson loops, i.e. $\xi_{\mathrm{rec}} \asymp f(\sigma)$ for a model- and geometry-dependent function f. In $3+1$ spacetime dimensions, dimensional analysis suggests $f(\sigma) \propto \sigma^{-1/2}$ (at fixed lattice spacing).

Remark 7.4 (Dimensional analysis). In $d+1$ spacetime dimensions, the string tension has dimensions $[\sigma] = (\mathrm{length})^{-(d-1)}$. At fixed lattice spacing, one may compare ξ_{rec} to $\sigma^{-1/(d-1)}$ as a natural length scale. The conjecture does not claim a universal proportionality constant.

7.1 Operational predictions

Fix ε and compute $d_{\mathrm{rec}}(\varepsilon)$ as in Definition 4.5. The conjectures predict:

- $d_{\mathrm{rec}}(\varepsilon)$ increases near critical points (diverging correlation length).
- In confining regimes, ξ_{rec} or $d_{\mathrm{rec}}(\varepsilon)$ correlates with σ_{eff} extracted from Wilson-loop data, up to finite-size effects and subsystem-prescription effects.

8 Numerical protocol

1. Choose lattice $\mathcal{L}$, model parameters, and a state ρ (ground state by ED/tensor network, or thermal state $\rho_\beta \propto e^{-\beta H}$).
2. Fix a subsystem prescription (default: EHS), including edge-mode conventions and a map ι if needed.
3. Choose a family of tripartitions $(A, B(w), C)$ parameterized by collar width w.
4. Compute reduced states ρ_{ABC}, ρ_{AB}, ρ_B, ρ_{BC} in the same prescription.
5. Choose a Petz regularization rule (Definition 4.1) and compute $E_{\mathrm{rec}}^{\mathrm{Petz}}(w)$ (Definition 4.4).
6. Extract ξ_{rec} (Definition 4.6) and/or $d_{\mathrm{rec}}(\varepsilon)$ (Definition 4.5).
7. Compute Wilson loops and extract σ_{eff} (area-law fits and/or Creutz ratios).
8. Compare the resulting scales and perform finite-size scaling checks.

9 Control experiment: TFIM validation

We validate the numerical pipeline (reduced states → Petz-type reconstruction → normalization → fidelity) on the one-dimensional transverse-field Ising model (TFIM) as a control system. The TFIM Hamiltonian with open boundary conditions is

$$H = -J \sum_{i=1}^{L-1} Z_i Z_{i+1} - g \sum_{i=1}^{L} X_i - h \sum_{i=1}^{L} Z_i.$$

For $g < 1$ and finite L, the symmetric ground state can be cat-like; we therefore include a small longitudinal field $h = 10^{-3}$ to select a symmetry-broken vacuum for the ordered regime. We choose contiguous tripartitions A–$B(w)$–C and trace out the remaining degrees of freedom as an environment D so that $|A|$ and $|C|$ are fixed while w varies.

9.1 Representative fit results

For parameters $L = 14$, $|A| = 3$, $|C| = 4$, and a semilog fit of $\log E_{\mathrm{rec}}^{\mathrm{Petz}}(w)$ over $w \in \{2, 3, 4\}$, we obtain the representative values shown in Table 1.

g	h	fitted ξ_{rec}	R^2 (semilog fit)
0.5	10^{-3}	1.424	0.6712
1.0	0	4.58	0.9276
2.0	0	0.6515	0.9990

Table 1: TFIM control experiment: recoverability-rate fits on $w \in \{2, 3, 4\}$ for contiguous A–$B(w)$–C with the complement traced out. The ordered regime uses a small symmetry-breaking field $h = 10^{-3}$ to avoid a finite-size cat state. In this experiment the median trace of the unnormalized reconstructed operator is numerically equal to 1 to the displayed precision; slight variations in the ordered regime fit parameters may occur across platforms due to solver-level numerical variability.

Remark 9.1 (Fit conventions). For the TFIM data in Table 1, we use the global regularization $\delta = 10^{-12}$ and perform semilog fits over the window $w \in \{2, 3, 4\}$, including only points with $E_{\mathrm{rec}}^{\mathrm{Petz}}(w) > 10^{-14}$ to avoid numerical-floor effects.

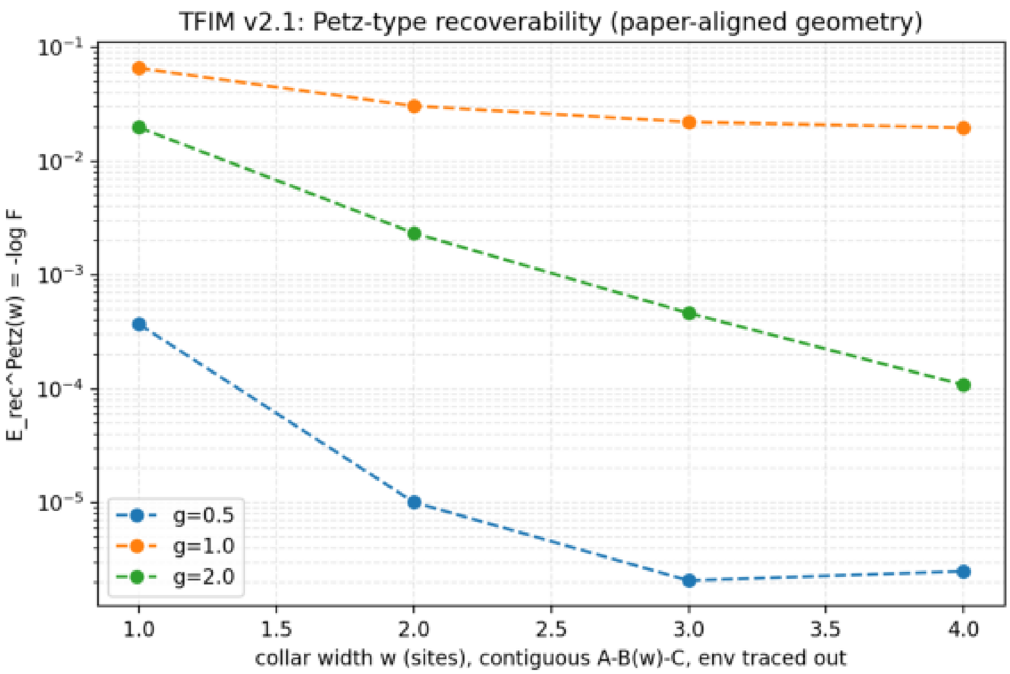

Figure 1: TFIM control experiment. Petz-type recoverability error $E_{\mathrm{rec}}^{\mathrm{Petz}}(w)$ versus collar width w for contiguous tripartitions A–$B(w)$–C with $|A|$ and $|C|$ fixed and the complement traced out. A small longitudinal field $h = 10^{-3}$ is used for $g < 1$ to avoid a finite-size cat state. The semilog plot exhibits approximately linear decay in gapped regimes and slower decay near the critical point.

10 Discussion and outlook

Relation to [1]. The companion note *Petz recoverability in AQFT via conditional expectations: a framework and a conditional exponential recovery bound* [1] isolates the algebraic ingredients needed to formulate Petz-type recoverability in type III AQFT under explicit bridge assumptions. The present paper proposes lattice computations that may provide evidence for analogous bridge behavior in gauge theories.

Open issues. A fully algebraic (gauge-invariant) implementation of $E_{\mathrm{rec}}^{\mathrm{Petz}}(w)$, including fidelity and recovery in the presence of centers and superselection sectors, remains an important direction for future work. A second immediate direction is to implement the full gauge testbed described in Section 6 and compare extracted ξ_{rec} to Wilson-loop string-tension proxies.

A Reproducibility: minimal TFIM script

This appendix provides a minimal Python script reproducing the TFIM control experiment reported in Section 9. It computes the TFIM ground state by sparse exact diagonalization, constructs reduced states for contiguous tripartitions A–$B(w)$–C with the complement traced out, applies the regularized Petz-type reconstruction, evaluates $E_{\mathrm{rec}}^{\mathrm{Petz}}(w)$, and performs a semilog fit to extract ξ_{rec}.

Dependencies. Python 3 with `numpy`, `scipy`, and `matplotlib`.

Run. Execute `python tfim_reproduce.py`. The script prints the ground-state energies and fit diagnostics and produces a semilog plot of $E_{\mathrm{rec}}^{\mathrm{Petz}}(w)$ versus w.

Listing 1: tfim_reproduce.py (minimal reproducible script)

```
import numpy as np
import scipy.linalg as la
import scipy.sparse as sp
import scipy.sparse.linalg as spla
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt

# -------------------------
# Linear algebra utilities
# -------------------------
def sqrtm_psd(M, eps=0.0):
    w, V = la.eigh(np.asarray(M))
    w = np.maximum(w, eps)
    return (V * np.sqrt(w)) @ V.conj().T

def invsqrtm_pd(M):
    w, V = la.eigh(np.asarray(M))
    if np.min(w) <= 0:
        raise ValueError("Matrix␣not␣strictly␣positive;␣increase␣delta.")
    return (V * (1.0/np.sqrt(w))) @ V.conj().T

def fidelity_squared(rho, sigma):
    sr = sqrtm_psd(rho)
    inner = sr @ sigma @ sr
    return (np.trace(sqrtm_psd(inner)).real)**2
```

```
# -------------------------
# Partial trace utilities
# -------------------------
def reduce_pure_to_density(psi, dims, keep):
    dims = list(dims)
    N = len(dims)
    keep = list(keep)
    trace = [i for i in range(N) if i not in keep]

    psi_t = psi.reshape(dims)
    perm = keep + trace
    psi_perm = np.transpose(psi_t, axes=perm)

    d_keep = int(np.prod([dims[i] for i in keep]))
    d_tr = int(np.prod([dims[i] for i in trace]))

    M = psi_perm.reshape(d_keep, d_tr)
    rho_keep = M @ M.conj().T
    return 0.5*(rho_keep + rho_keep.conj().T)

def partial_trace_rho(rho, dims, keep):
    dims = list(dims)
    N = len(dims)
    keep = list(keep)
    trace = [i for i in range(N) if i not in keep]

    d_keep = int(np.prod([dims[i] for i in keep]))
    d_tr = int(np.prod([dims[i] for i in trace]))

    rho = np.asarray(rho, dtype=np.complex128)
    rho_t = rho.reshape(dims + dims)

    perm = keep + trace + [i+N for i in keep] + [i+N for i in trace]
    rho_p = np.transpose(rho_t, axes=perm)
    rho_p = rho_p.reshape(d_keep, d_tr, d_keep, d_tr)

    rho_keep = np.trace(rho_p, axis1=1, axis2=3)
    return 0.5*(rho_keep + rho_keep.conj().T)

# -------------------------
# TFIM (open boundary conditions)
# H = -J sum Z_i Z_{i+1} - g sum X_i - h sum Z_i
# -------------------------
def pauli():
    X = sp.csr_matrix(np.array([[0,1],[1,0]], dtype=np.float64))
    Z = sp.csr_matrix(np.array([[1,0],[0,-1]], dtype=np.float64))
    I = sp.identity(2, format="csr", dtype=np.float64)
    return X, Z, I

def op_on_site(op, i, L):
    X, Z, I = pauli()
    out = None
    for k in range(L):
        factor = op if k == i else I
        out = factor if out is None else sp.kron(out, factor, format="csr")
    return out

def tfim_hamiltonian(L, J=1.0, g=1.0, h=0.0):
```

```python
    X, Z, I = pauli()
    dim = 2**L
    H = sp.csr_matrix((dim, dim), dtype=np.float64)

    for i in range(L-1):
        Zi = op_on_site(Z, i, L)
        Zip1 = op_on_site(Z, i+1, L)
        H = H - J*(Zi @ Zip1)

    for i in range(L):
        Xi = op_on_site(X, i, L)
        H = H - g*Xi

    if h != 0.0:
        for i in range(L):
            Zi = op_on_site(Z, i, L)
            H = H - h*Zi

    return H

def ground_state_tfim(L, J=1.0, g=1.0, h=0.0):
    H = tfim_hamiltonian(L, J=J, g=g, h=h)
    vals, vecs = spla.eigsh(H, k=1, which="SA", tol=1e-10, maxiter=12000)
    psi = vecs[:,0].astype(np.complex128)
    psi /= np.linalg.norm(psi)
    return float(vals[0]), psi

# -------------------------
# Geometry: contiguous A-B(w)-C in a chain; trace the complement as environment
# -------------------------
def rho_ABC_from_chain_pure(psi_full, L, LA, w, LC, offset=0):
    startA = offset
    startB = startA + LA
    startC = startB + w
    endC = startC + LC
    if endC > L:
        raise ValueError("Need offset + LA + w + LC <= L.")

    A_sites = list(range(startA, startA + LA))
    B_sites = list(range(startB, startB + w))
    C_sites = list(range(startC, endC))
    keep_sites = A_sites + B_sites + C_sites

    rho_ABC = reduce_pure_to_density(psi_full, [2]*L, keep=keep_sites)
    return rho_ABC, (2**LA, 2**w, 2**LC)

# -------------------------
# Petz-type reconstruction
# -------------------------
def petz_recover_mixed(rho_ABC, dims_ABC, delta=1e-12, renormalize=True):
    dA, dB, dC = dims_ABC
    dims = [dA, dB, dC]

    rho_AB = partial_trace_rho(rho_ABC, dims, keep=[0,1])
    rho_B = partial_trace_rho(rho_ABC, dims, keep=[1])
    rho_BC = partial_trace_rho(rho_ABC, dims, keep=[1,2])

    rho_B_delta = rho_B + delta*np.eye(dB, dtype=np.complex128)
    O_B = invsqrtm_pd(rho_B_delta)
```

```
    sqrt_rho_BC = sqrtm_psd(rho_BC)

    IA = np.eye(dA, dtype=np.complex128)
    IC = np.eye(dC, dtype=np.complex128)

    sandwich = np.kron(IA, O_B)
    middle = sandwich @ rho_AB @ sandwich
    middle = 0.5*(middle + middle.conj().T)

    left = np.kron(IA, sqrt_rho_BC)
    sigma = left @ np.kron(middle, IC) @ left
    sigma = 0.5*(sigma + sigma.conj().T)

    tr_sigma = float(np.trace(sigma).real)
    if renormalize:
        if tr_sigma <= 0:
            raise ValueError("tr(sigma) <= 0; increase delta.")
        sigmaN = sigma / tr_sigma
    else:
        sigmaN = sigma

    F = float(np.clip(fidelity_squared(rho_ABC, sigmaN), 0.0, 1.0))
    E = -np.log(F + 1e-16)
    return E, F, tr_sigma

def fit_xi_with_r2(w, E, wmin=2, Emin=1e-14):
    w = np.asarray(w); E = np.asarray(E)
    mask = (w >= wmin) & (E > Emin)
    x = w[mask].astype(float)
    y = np.log(E[mask].astype(float))

    n = len(x)
    if n < 2:
        return np.nan, n, np.nan

    A = np.vstack([np.ones_like(x), -x]).T
    coeff, *_ = la.lstsq(A, y)
    inv_xi = coeff[1]
    yhat = A @ coeff

    ss_res = float(np.sum((y - yhat)**2))
    ss_tot = float(np.sum((y - np.mean(y))**2))
    r2 = 1.0 - ss_res/ss_tot if ss_tot > 0 else np.nan

    xi = np.inf if inv_xi <= 0 else 1.0/inv_xi
    return xi, n, r2

def run():
    L, LA, LC = 14, 3, 4
    J, delta = 1.0, 1e-12
    w_cap = 4
    g_list = (0.5, 1.0, 2.0)
    h_small = 1e-3

    results = {}
    for g in g_list:
        h = h_small if g < 1.0 else 0.0
        E0, psi = ground_state_tfim(L, J=J, g=g, h=h)
        print(f"g={g:>4}  h={h:.1e}  E0={E0:.10f}")
```

```python

        wmax = min(L - LA - LC, w_cap)
        ws, Es, Fs, trs = [], [], [], []
        for w in range(1, wmax+1):
            block = LA + w + LC
            offset = (L - block)//2
            dim_ABC = 2**block
            print(f"  w={w}  dim(ABC)={dim_ABC}  offset={offset}")

            rho_ABC, dims_ABC = rho_ABC_from_chain_pure(psi, L, LA, w, LC, offset=offset)
            Erec, F, tr_sigma = petz_recover_mixed(rho_ABC, dims_ABC, delta=delta, renormalize=True)
            ws.append(w); Es.append(Erec); Fs.append(F); trs.append(tr_sigma)
        results[g] = (np.array(ws), np.array(Es), np.array(Fs), np.array(trs))

    # plot
    plt.figure(figsize=(8,5))
    for g, (w, E, F, tr) in results.items():
        plt.plot(w, E, "o--", label=f"g={g}")
    plt.yscale("log")
    plt.xlabel("collar width w (sites), contiguous A-B(w)-C, env traced out")
    plt.ylabel("E_rec^Petz(w) = -log F")
    plt.title("TFIM v2.1: Petz-type recoverability (paper-aligned geometry)")
    plt.grid(True, which="both", ls="--", alpha=0.3)
    plt.legend()
    plt.savefig("tfim_v21_reproduced.png", dpi=200, bbox_inches="tight")
    print("Saved figure: tfim_v21_reproduced.png")

    print("Fit window: w>=2 (semilog fit of log(E) vs w)")
    for g, (w, E, F, tr) in results.items():
        xi, n, r2 = fit_xi_with_r2(w, E, wmin=2, Emin=1e-14)
        print(f"g={g}: xi_rec  {xi:.4g}   (used n={n} points)   R^2={r2:.4f}   median tr(sigma)={np.median(tr):.6f}")

if __name__ == "__main__":
    run()
```

Notes. The script is intended for transparency and validation on small system sizes. For larger systems one should use tensor-network methods rather than dense matrix square roots.

References

[1] L. Eriksson, Petz recoverability in AQFT via conditional expectations: a framework and a conditional exponential recovery bound, preprint (2026).

[2] D. Petz, Sufficient subalgebras and the relative entropy of states of a von Neumann algebra, *Communications in Mathematical Physics* **105**, 123–131 (1986).

[3] O. Fawzi and R. Renner, Quantum conditional mutual information and approximate Markov chains, *Communications in Mathematical Physics* **340**, 575–611 (2015).

[4] J. B. Kogut, An introduction to lattice gauge theory and spin systems, *Reviews of Modern Physics* **51**, 659–713 (1979).

[5] W. Donnelly, Decomposition of entanglement entropy in lattice gauge theory, *Physical Review D* **85**, 085004 (2012). arXiv:1109.0036.

[6] H. Casini, M. Huerta, and J. A. Rosabal, Remarks on entanglement entropy for gauge fields, *Physical Review D* **89**, 085012 (2014). arXiv:1312.1183.

[7] W. Donnelly, Entanglement entropy and non-Abelian gauge symmetry, *Classical and Quantum Gravity* **31**, 214003 (2014). arXiv:1406.7304.

[8] W. Donnelly and A. C. Wall, Entanglement entropy of electromagnetic edge modes, *Physical Review Letters* **114**, 111603 (2015). arXiv:1412.1895.

[9] S. Aoki, T. Iritani, M. Nozaki, T. Numasawa, N. Shiba, and H. Tasaki, On the definition of entanglement entropy in lattice gauge theories, *Journal of High Energy Physics* **06**, 187 (2015). arXiv:1502.04267.

[10] R. M. Soni and S. P. Trivedi, Aspects of entanglement entropy for gauge theories, *Journal of High Energy Physics* **01**, 136 (2016). arXiv:1510.07455.