Finite-Dimensional Davies Interface Lemmas and TFIM Witness Tests for the Heisenberg Cut as a Resource Boundary

Lluis Eriksson

December 2025

Status of claims (read first)

- **Imported result.** The maintenance inequality $P_{\text{extra}}(\rho) \geq k_B T\,\dot{C}_{\text{loss}}(\rho)$ is imported from the published preprint [2].
- **Proved here (finite-dimensional).** The exact $\omega = 0$ Dirichlet identity (Lemma 1), the Bohr-block Dirichlet decomposition (Lemma 2), and the witness lower bound (Proposition 1).
- **Sufficient conditional bound.** The envelope suppression lemma (Lemma 3) is a finite-dimensional sufficient statement under explicit hypotheses (IR exclusion + quasi-local spectral tails).
- **Numerics.** TFIM results are exact diagonalization and stability-checked finite-size diagnostics (here $N \leq 12$); no thermodynamic-limit uniformity is claimed.

Abstract

We present a finite-dimensional technical core for interpreting the quantum–classical boundary as a control-resource limitation: a state is operationally "classical" (relative to a chosen conditional expectation) whenever sustaining coherence is infeasible under a finite available power budget.

Using battery-assisted thermal operations at temperature T and energy pinching Δ, we quantify coherence by $C(\rho) = S(\rho\|\Delta[\rho])$ and define its instantaneous loss rate $\dot{C}_{\text{loss}}(\rho)$ under uncontrolled Markovian dynamics. An imported maintenance inequality yields the operational bound $P_{\text{extra}}(\rho) \geq k_B T\,\dot{C}_{\text{loss}}(\rho)$.

We isolate the dynamical hinge required for geometric scaling claims: the relation between geometric separation and effective decoherence rates. In Davies generators we give two interface lemmas: a negative lemma showing that an active $\omega = 0$ channel yields a witness-based lower bound via a KMS commutator with $S(0)$, and a positive sufficient lemma yielding envelope suppression under explicit infrared exclusion plus a quasi-local spectral tail hypothesis.

Finally, we provide TFIM exact-diagonalization witness protocols computing ratios $R(\epsilon)$ and optimized local witnesses $R_{\text{opt}}(\epsilon)$, including scaling tests with $\epsilon = \epsilon(N)$ and temperature sweeps over β. We also report direct state-side interface micro-tests on weak-coherence families, confirming plateau behavior of $\dot{C}_{\text{loss}}/C$ and controlled $\omega = 0$ scaling in $\gamma(0)$ and β, with stability checks in the Euler step size and PSD clipping threshold.

Contents

1 Scope and non-claims

This work does not propose modified quantum dynamics, collapse mechanisms, derivations of the Born rule, or ontological resolutions of the measurement problem. The claims are operational: they concern minimal control resources required to maintain coherence against uncontrolled open-system dynamics under an explicit thermodynamic bookkeeping model.

Remark 1 (Figures)**.** *This manuscript expects several figure files (e.g.* `scaling_Ropt_N_over_4.png`*,* `interface_ratio_vs_C.png`*) to be present in 'figures/' (or the working directory). If they are missing, compilation will fail.*

Table 1: Program Map: logical structure of the coherence-maintenance series.

ID	Role	Key Contribution
	Foundational Preprints (Published)	
P1	Geometric Foundation	AQFT static clustering bounds and Petz recovery.
P2	Thermodynamic Law	Maintenance inequality: $P_{\mathrm{extra}}(\rho) \geq k_{\mathrm{B}}T\,\dot{C}_{\mathrm{loss}}(\rho)$.
P3	Synthesis/Outlook	Operational cut framing and interface discussion.
	Core Technical Paper (This Work)	
P7	**Finite-Dimensional Core**	**Davies interface lemmas, TFIM witnesses, and reproducible micro-tests.**
	Companion Notes (to be posted separately)	
P4	Hypothesis Framing	RIP: weak vs. strong forms, plausibility routes, and falsification criteria.
P6	Stress Tests	Operatorial $\kappa(\epsilon)$ tests and $\omega = 0$-floor mechanisms.
P5	Logical Closure	Separation of proved results vs. conditional interfaces vs. open dynamical gaps.

2 Finite-dimensional operational core

2.1 Thermal state, energy pinching, and coherence

Let the system Hamiltonian be

$$H_S = \sum_n E_n \Pi_n,$$

and define at temperature T the thermal state

$$\gamma_S := \frac{e^{-\beta H_S}}{\mathrm{Tr}(e^{-\beta H_S})}, \qquad \beta = \frac{1}{k_{\mathrm{B}}T}.$$

Define energy pinching

$$\Delta[\rho] := \sum_n \Pi_n \rho \Pi_n,$$

and relative-entropy coherence (nats)

$$C(\rho) := S(\rho \| \Delta[\rho]), \qquad S(\rho\|\sigma) := \mathrm{Tr}[\rho(\log\rho - \log\sigma)]\,.$$

2.2 Uncontrolled Markovian dynamics and coherence-loss rate

Let $\rho_t = e^{t\mathcal{L}}(\rho)$ be a Markovian semigroup with a thermal stationary state $\mathcal{L}(\gamma_S) = 0$. Define the instantaneous coherence-loss rate

$$\dot{C}_{\text{loss}}(\rho) := -\left.\frac{d}{dt}C(\rho_t)\right|_{t=0}.$$

2.3 Control model and extra power

We assume the control primitive of battery-assisted thermal operations at temperature T, and define incremental (extra) power P_{extra} as an infimum over pairs of strategies maintaining ρ and maintaining $\Delta[\rho]$, respectively.

2.4 Core maintenance inequality (imported)

Theorem 1 (Extra-power coherence maintenance bound (imported))**.** *Assume strategies exist that maintain ρ and that maintain $\Delta[\rho]$ under battery-assisted thermal operations at temperature T. Then*

$$P_{\text{extra}}(\rho) \geq k_{\mathrm{B}}T\,\dot{C}_{\text{loss}}(\rho).$$

Corollary 1 (No-go from a uniform loss floor)**.** *Fix $C_0 > 0$ and a nonempty target family $\mathcal{F}_{C_0} \subset \{\rho : C(\rho) \geq C_0\}$. If there exists*

$$g(C_0) := \inf_{\rho \in \mathcal{F}_{C_0}} \dot{C}_{\text{loss}}(\rho) > 0,$$

then any controller maintaining any $\rho \in \mathcal{F}_{C_0}$ requires $P_{\text{extra}}(\rho) \geq k_{\mathrm{B}}T\,g(C_0)$. If $P_{\max} < k_{\mathrm{B}}T\,g(C_0)$, sustained maintenance is operationally impossible on $\mathcal{F}_{C_0}$.

3 Davies generators, KMS Dirichlet form, and an effective envelope

3.1 Bohr components

Consider a Davies generator derived in the weak-coupling–secular limit with a single system coupling operator $S = S^\dagger$. Define Bohr components

$$S(\omega) := \sum_{E_m - E_n = \omega} \Pi_m S \Pi_n, \qquad S = \sum_{\omega} S(\omega), \qquad S(\omega)^\dagger = S(-\omega).$$

Bath rates satisfy the KMS relation

$$\gamma(-\omega) = e^{-\beta\omega}\gamma(\omega).$$

3.2 KMS inner product and Dirichlet form

Define the KMS (GNS) inner product

$$\langle A, B\rangle_\sigma := \mathrm{Tr}\Big(\sigma^{1/2}A^\dagger\sigma^{1/2}B\Big), \qquad \|A\|_{2,\sigma}^2 := \langle A, A\rangle_\sigma,$$

with $\sigma = \gamma_S$. For the Heisenberg-picture generator $\mathcal{L}^\dagger$, define the Dirichlet form

$$\mathcal{E}_\sigma(O) := -\mathrm{Re}\,\langle O, \mathcal{L}^\dagger(O)\rangle_\sigma.$$

3.3 Separation and variational envelope

Fix a coupling region (e.g. a site j_0) and let $\mathcal{A}_\epsilon$ denote operators supported at distance at least ϵ from that region. Define the operatorial envelope

$$\kappa(\epsilon) := \sup_{O \in \mathcal{A}_\epsilon,\ O \neq 0} \frac{\mathcal{E}_\sigma(O)}{\|O\|_{2,\sigma}^2}.$$

3.4 A concrete tail map on spin chains (Pauli-string conditional expectation)

In lattice spin systems we use a canonical "tail" map built from the Pauli-string expansion. Let Λ be the set of lattice sites and let $\mathcal{A}(\Omega)$ denote the local operator algebra on region $\Omega \subset \Lambda$.

Fix a coupling region Ω_0 (e.g. $\Omega_0 = \{j_0\}$). For $\epsilon \geq 1$ define the far region

$$\Omega_\epsilon := \{ j \in \Lambda :\ d(j, \Omega_0) \geq \epsilon \}.$$

Let $\mathcal{A}_{<\epsilon} := \mathcal{A}(\Lambda \setminus \Omega_\epsilon)$ be the algebra supported within distance $< \epsilon$ of the coupling region.

Define the *Pauli-string conditional expectation* $\mathbb{E}_{<\epsilon} : \mathcal{A}(\Lambda) \to \mathcal{A}_{<\epsilon}$ as the orthogonal projection (with respect to the normalized Hilbert–Schmidt inner product) onto the span of Pauli strings supported in $\Lambda \setminus \Omega_\epsilon$. Equivalently, writing any operator as a Pauli expansion $X = \sum_P x_P P$ with Pauli strings P,

$$\mathbb{E}_{<\epsilon}[X] := \sum_{P:\ \mathrm{supp}(P) \subseteq \Lambda \setminus \Omega_\epsilon} x_P \, P.$$

We then define the tail at separation ϵ by

$$\mathrm{tail}_\epsilon(X) := X - \mathbb{E}_{<\epsilon}[X].$$

Remark 2. *We define $\mathbb{E}_{<\epsilon}$ as a normalized Hilbert–Schmidt projection for concreteness and computational convenience. Since the system is finite-dimensional, $\|\cdot\|_{2,\sigma}$ and the Hilbert–Schmidt norm are equivalent; thus any exponential tail bound in one induces an exponential tail bound in the other up to multiplicative constants depending on σ.*

These constants may depend on (N, β) in finite-size tests. Alternatively, one may define $\mathbb{E}_{<\epsilon}$ as the orthogonal projection with respect to $\langle \cdot, \cdot \rangle_\sigma$ to align norms more tightly.

4 Interface lemmas: rate floors and conditional envelope suppression

4.1 Negative lemma: the $\omega = 0$ channel yields a witness lower bound

Define the $\omega = 0$ Bohr component

$$S(0) := \sum_n \Pi_n S \Pi_n.$$

Assume $\gamma(0) > 0$ (e.g. after infrared regularization in an Ohmic model).

Lemma 1 (Exact $\omega = 0$ Dirichlet identity)**.** *If the $\omega = 0$ Davies channel is*

$$\mathcal{L}_0^\dagger(O) = \gamma(0) \left(S(0) O S(0) - \frac{1}{2} \{ S(0)^2, O \} \right),$$

then for all O,

$$\mathcal{E}_\sigma^{(0)}(O) := -\mathrm{Re}\, \langle O, \mathcal{L}_0^\dagger(O) \rangle_\sigma = \frac{\gamma(0)}{2} \, \|[S(0), O]\|_{2,\sigma}^2.$$

Proof. Let $\sigma = \gamma_S$ and set $S_0 := S(0)$. Since S_0 is diagonal in the energy eigenbasis, we have $[S_0, H_S] = 0$ and hence $[S_0, \sigma] = 0$, i.e. $\sigma^{1/2} S_0 = S_0 \sigma^{1/2}$.

Write

$$\mathcal{L}_0^\dagger(O) = \gamma(0)\left(S_0 O S_0 - \frac{1}{2}\{S_0^2, O\}\right).$$

Using $\langle A, B\rangle_\sigma = \mathrm{Tr}(\sigma^{1/2} A^\dagger \sigma^{1/2} B)$ and $[S_0, \sigma^{1/2}] = 0$, one expands both $\langle O, \mathcal{L}_0^\dagger(O)\rangle_\sigma$ and $\|[S_0, O]\|_{2,\sigma}^2$ and matches terms, obtaining

$$-\mathrm{Re}\,\langle O, \mathcal{L}_0^\dagger(O)\rangle_\sigma = \frac{\gamma(0)}{2}\,\|[S_0, O]\|_{2,\sigma}^2.$$

□

Proposition 1 (Fixed-separation witness lower bound (sufficient)). *Fix $\epsilon_0 \geq 1$. If there exists $c_{\epsilon_0} > 0$ and an operator $O_{\epsilon_0} \in \mathcal{A}_{\epsilon_0}$ such that*

$$\frac{\|[S(0), O_{\epsilon_0}]\|_{2,\sigma}^2}{\|O_{\epsilon_0}\|_{2,\sigma}^2} \geq c_{\epsilon_0},$$

then

$$\kappa(\epsilon_0) \geq \frac{\gamma(0)}{2}\, c_{\epsilon_0}.$$

4.2 Bohr-block decomposition of the KMS Dirichlet form

Lemma 2 (Bohr-block decomposition of the KMS Dirichlet form (single-channel Davies)). *Assume a single Hermitian coupling operator $S = S^\dagger$ and a Davies generator in the Heisenberg picture of the form*

$$\mathcal{L}^\dagger(O) = \sum_\omega \gamma(\omega)\left(S(\omega)^\dagger O S(\omega) - \frac{1}{2}\{S(\omega)^\dagger S(\omega), O\}\right), \qquad S(\omega)^\dagger = S(-\omega),$$

with rates satisfying the KMS relation $\gamma(-\omega) = e^{-\beta\omega}\gamma(\omega)$ and reference state $\sigma = \gamma_S$. Let

$$\langle A, B\rangle_\sigma := \mathrm{Tr}\left(\sigma^{1/2} A^\dagger \sigma^{1/2} B\right), \qquad \mathcal{E}_\sigma(O) := -\mathrm{Re}\,\langle O, \mathcal{L}^\dagger(O)\rangle_\sigma.$$

Then

$$\mathcal{E}_\sigma(O) = \frac{1}{2}\sum_\omega \gamma(\omega)\,\|[S(\omega), O]\|_{2,\sigma}^2.$$

Proof. Fix ω and set $L := S(\omega)$ and $\gamma := \gamma(\omega)$. Define

$$\mathcal{L}_\omega^\dagger(O) := \gamma\left(L^\dagger O L - \frac{1}{2}\{L^\dagger L, O\}\right).$$

Since $\sigma = \gamma_S \propto e^{-\beta H_S}$ and L has Bohr frequency ω, the intertwining relations hold:

$$\sigma^{1/2} L = e^{-\beta\omega/2} L \sigma^{1/2}, \qquad \sigma^{1/2} L^\dagger = e^{+\beta\omega/2} L^\dagger \sigma^{1/2}.$$

Using these relations and cyclicity of trace, one rewrites $-\mathrm{Re}\,\langle O, \mathcal{L}_\omega^\dagger(O)\rangle_\sigma$ as $\frac{\gamma}{2}\|[L, O]\|_{2,\sigma}^2$. Summing over ω gives the claim. □

4.3 Positive lemma: envelope suppression under IR exclusion and quasi-local spectral tails

Definition 1 (Infrared exclusion)**.** *We say the coupling satisfies IR exclusion at scale* $\omega_* > 0$ *if* $S(\omega) = 0$ *for all* $|\omega| < \omega_*$ *(or alternatively* $\gamma(\omega) = 0$ *on that band).*

Lemma 3 (Envelope suppression under IR exclusion and quasi-local spectral tails (sufficient))**.** *Assume IR exclusion and suppose there exist* $A > 0$*,* $\xi > 0$*, and* $p \geq 0$ *such that for all* $|\omega| \geq \omega_*$*,*

$$\|\mathrm{tail}_\epsilon(S(\omega))\|_{2,\sigma} \leq A(1+\epsilon)^p e^{-\epsilon/\xi},$$

and $\sup_{|\omega|\geq\omega_*} \gamma(\omega) < \infty$*. Then there exists* $C > 0$ *such that*

$$\kappa(\epsilon) \leq C(1+\epsilon)^{2p} e^{-2\epsilon/\xi}.$$

Proof (sketch). By Lemma 2,

$$\mathcal{E}_\sigma(O) = \frac{1}{2}\sum_\omega \gamma(\omega)\, \|[S(\omega), O]\|_{2,\sigma}^2.$$

By IR exclusion, only $|\omega| \geq \omega_*$ contribute. For $O \in \mathcal{A}_\epsilon$, write

$$S(\omega) = \mathbb{E}_{<\epsilon}[S(\omega)] + \mathrm{tail}_\epsilon(S(\omega)).$$

Since $\mathbb{E}_{<\epsilon}[S(\omega)]$ is supported within distance $< \epsilon$ while O is supported at distance at least ϵ,

$$[\mathbb{E}_{<\epsilon}[S(\omega)], O] = 0, \qquad \Rightarrow \qquad [S(\omega), O] = [\mathrm{tail}_\epsilon(S(\omega)), O].$$

Then

$$\|[A, O]\|_{2,\sigma} \leq \|AO\|_{2,\sigma} + \|OA\|_{2,\sigma} \leq 2\|A\|_\infty\, \|O\|_{2,\sigma},$$

and hence

$$\frac{\|[S(\omega), O]\|_{2,\sigma}^2}{\|O\|_{2,\sigma}^2} \leq 4\, \|\mathrm{tail}_\epsilon(S(\omega))\|_\infty^2.$$

Using $\|X\|_\infty \leq \|X\|_2 \leq \lambda_{\min}(\sigma)^{-1/2}\|X\|_{2,\sigma}$ yields

$$\|\mathrm{tail}_\epsilon(S(\omega))\|_\infty^2 \leq \frac{1}{\lambda_{\min}(\sigma)}\, \|\mathrm{tail}_\epsilon(S(\omega))\|_{2,\sigma}^2.$$

Combining the bounded-rate assumption with the tail hypothesis gives the stated envelope for $\kappa(\epsilon)$ after taking the supremum over $O \in \mathcal{A}_\epsilon$. □

Remark 3 (Finite-size scaling of the prefactor)**.** *The constant* C *contains a factor* $\lambda_{\min}(\sigma)^{-1}$ *arising from norm comparisons and may grow with* (N, β) *since* $\sigma = \gamma_S \propto e^{-\beta H_S}$*. Accordingly, Lemma 3 is stated as a sufficient finite-dimensional bound at fixed* (N, β) *and is not claimed to be uniform in any thermodynamic limit.*

5 TFIM witness computations (exact diagonalization)

5.1 Model

We consider the transverse-field Ising model on N spins with open boundary conditions:

$$H_S = -J \sum_{i=1}^{N-1} \sigma_i^x \sigma_{i+1}^x - h \sum_{i=1}^{N} \sigma_i^z, \qquad h > J.$$

We choose a coupling site $j_0 = \lfloor N/2 \rfloor$ and take $S = \sigma_{j_0}^z$. We consider thermal states $\sigma = \gamma_S$ at inverse temperatures $\beta \in \{0.5, 1, 2\}$.

5.2 Witness ratios

For a distant site $k = j_0 \pm \epsilon$ define

$$R(\epsilon; O) := \frac{\|[S(0), O]\|_{2,\sigma}^2}{\|O\|_{2,\sigma}^2},$$

and define the optimized local witness ratio

$$R_{\mathrm{opt}}(\epsilon) := \max_{O \in \mathrm{span}\{\sigma_k^x, \sigma_k^y, \sigma_k^z\}} R(\epsilon; O).$$

By Lemma 1,

$$\kappa(\epsilon) \geq \frac{\gamma(0)}{2} R_{\mathrm{opt}}(\epsilon).$$

5.3 Fixed ϵ results (at $\beta = 1$)

For $J = 1$, $h = 1.5$, $\beta = 1$:

N	$R_{\mathrm{opt}}(1)$	$R_{\mathrm{opt}}(2)$	$R_{\mathrm{opt}}(3)$
8	0.134648	0.0885503	0.0684289
10	0.0999406	0.0696450	0.0550748
12	0.0793451	0.0574518	0.0464596

Table 2: Optimized local KMS commutator witness $R_{\mathrm{opt}}(\epsilon)$ for TFIM with $S = \sigma_{j_0}^z$, center coupling, $\beta = 1$, $h = 1.5$, $J = 1$.

5.4 Scaling tests with $\epsilon = \epsilon(N)$ (at $\beta = 1$)

We computed $R_{\mathrm{opt}}(\epsilon(N))$ for four rules:

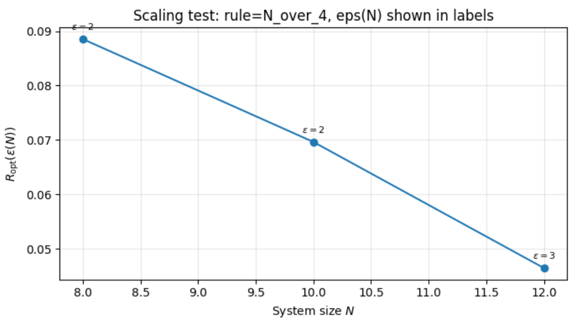
$$\epsilon(N) \in \left\{ \left\lfloor \frac{N}{4} \right\rfloor, \ \left\lfloor \frac{N}{3} \right\rfloor, \ \epsilon_{\max}(N), \ 3 \right\}, \qquad \epsilon_{\max}(N) = \min(j_0, N - 1 - j_0).$$

5.5 Temperature sweep tests over β

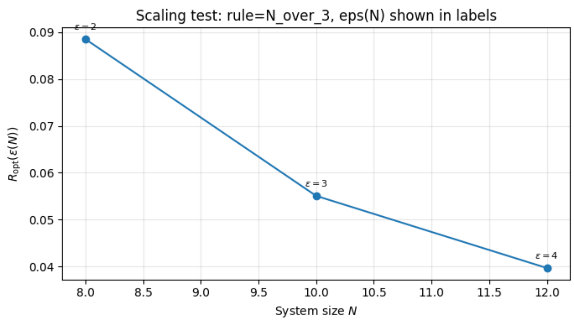
We swept $\beta \in \{0.5, 1, 2\}$ for the same $\epsilon(N)$ rules and $N \in \{8, 10, 12\}$.

Rule	$N = 8$	$N = 10$	$N = 12$
$\lfloor N/4 \rfloor$	0.0885503 ($\epsilon = 2$)	0.0696450 ($\epsilon = 2$)	0.0464596 ($\epsilon = 3$)
$\lfloor N/3 \rfloor$	0.0885503 ($\epsilon = 2$)	0.0550748 ($\epsilon = 3$)	0.0396591 ($\epsilon = 4$)
$\epsilon_{\max}(N)$	0.0684289 ($\epsilon = 3$)	0.0461050 ($\epsilon = 4$)	0.0330550 ($\epsilon = 5$)
3 (fixed)	0.0684289 ($\epsilon = 3$)	0.0550748 ($\epsilon = 3$)	0.0464596 ($\epsilon = 3$)

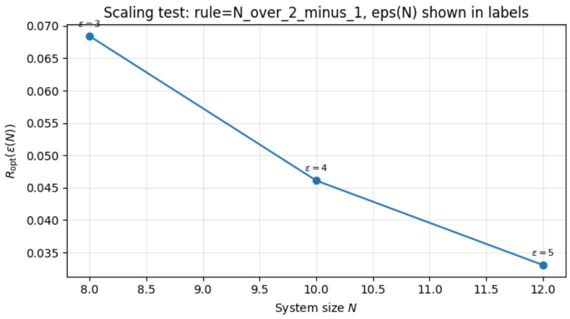
Table 3: Scaling test values $R_{\text{opt}}(\epsilon(N))$ for TFIM with $S = \sigma^z_{j_0}$, $\beta = 1$, $h = 1.5$, $J = 1$.



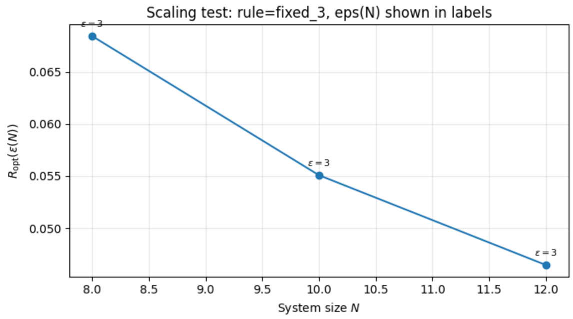
(a) $\epsilon = \lfloor N/4 \rfloor$



(b) $\epsilon = \lfloor N/3 \rfloor$

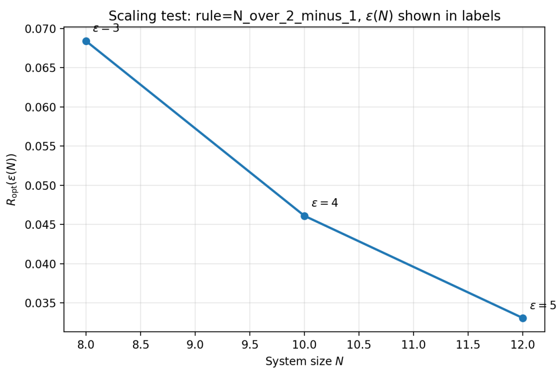


(c) $\epsilon = \epsilon_{\max}(N)$

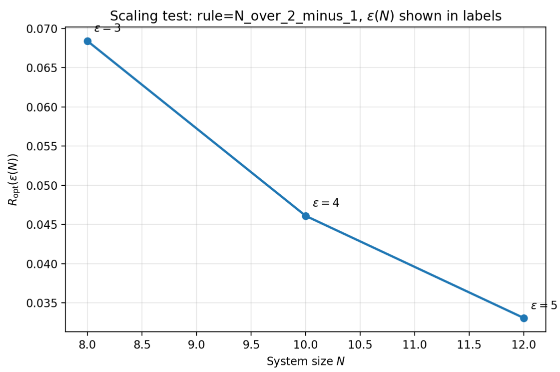


(d) $\epsilon = 3$

Figure 1: Scaling plots of $R_{\text{opt}}(\epsilon(N))$ under four $\epsilon(N)$ rules (at $\beta = 1$).

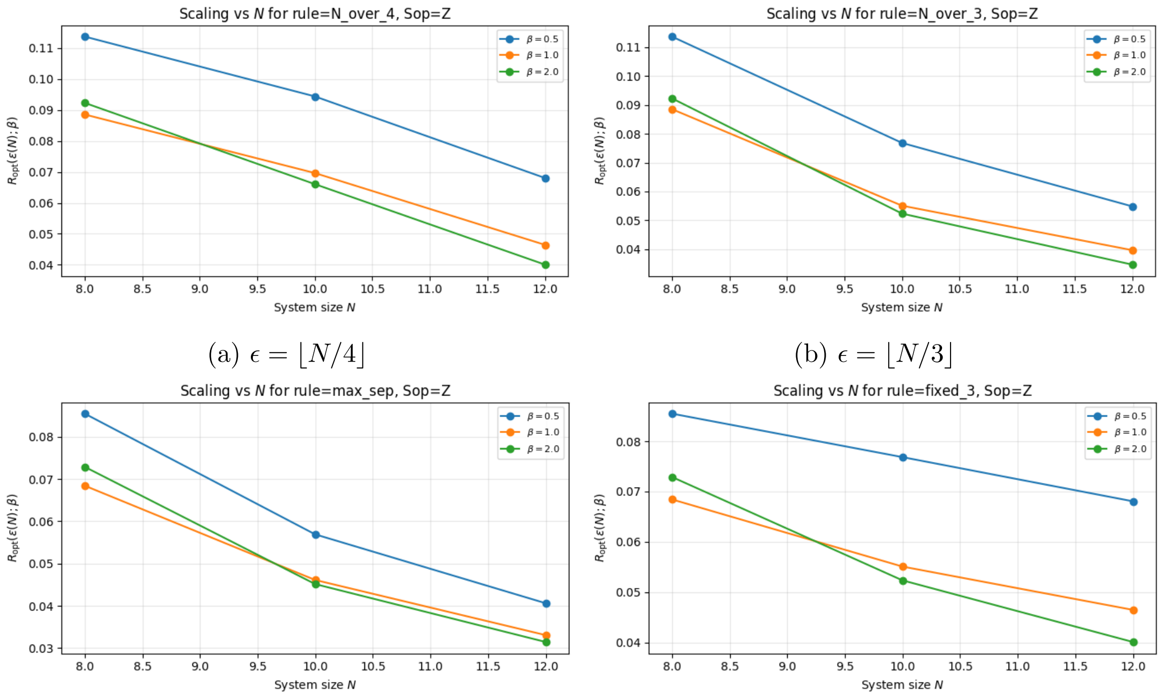


(a) Scaling test: rule=fixed_3, $\epsilon(N)$ shown in labels.



(b) Scaling test: rule=$\epsilon_{\max}(N)$, $\epsilon(N)$ shown in labels.

Figure 2: Scaling tests where $\epsilon(N)$ is printed on each data point.



(a) $\epsilon = \lfloor N/4 \rfloor$

(b) $\epsilon = \lfloor N/3 \rfloor$

(c) $\epsilon = \epsilon_{\max}(N)$

(d) $\epsilon = 3$

Figure 3: Optimized witness $R_{\mathrm{opt}}(\epsilon(N);\beta)$ versus N for multiple β values.

6 State-side interface micro-tests: weak-coherence families and $\omega = 0$ scaling

The witness computations above are observable-side tests based on the exact $\omega = 0$ Dirichlet identity. To directly probe the state-side quantity appearing in the maintenance inequality, we also test the interface numerically on explicit weak-coherence families.

Weak-coherence family. Fix an energy eigenbasis $\{n\}$ of H_S, thermal weights $p_n \propto e^{-\beta E_n}$, and a pair (m, n) with $m \neq n$. Define in the energy basis

$$\rho_E(\alpha) := \mathrm{diag}(p) + \alpha\sqrt{p_m p_n}\,(mn + nm)\,,$$

and set $\rho(\alpha) = V \rho_E(\alpha) V^\dagger$ where V diagonalizes H_S. For small α, this is positive and satisfies $\Delta[\rho(\alpha)] = \mathrm{diag}(p)$ by construction. We evaluate $C(\rho) = S(\rho \| \Delta[\rho])$ and approximate

$$\dot{C}_{\mathrm{loss}}(\rho) \approx -\frac{C(\rho_t) - C(\rho)}{t}, \qquad \rho_t \approx \Pi(\rho + t\,\mathcal{L}(\rho))\,,$$

where Π denotes numerical projection to a valid density matrix (Hermitian, PSD with eigenvalue clipping, trace one). We report the empirical ratio

$$r(\rho) := \frac{\dot{C}_{\mathrm{loss}}(\rho)}{C(\rho)}.$$

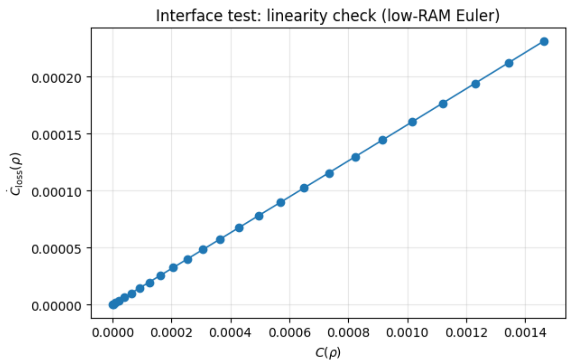
Observed plateau (truncated Davies blocks). For $N = 6$, $J = 1$, $h = 1.5$, $\beta = 1$, coupling $S = \sigma^z_{j_0}$ at the center site, and a transition-selected pair optimized for a nonzero Bohr block, we observed that $r(\rho)$ is approximately constant across several decades of $C(\rho)$ (excluding ultra-small C where PSD projection dominates), with a plateau around

$$r(\rho) \approx 0.157\text{–}0.158$$

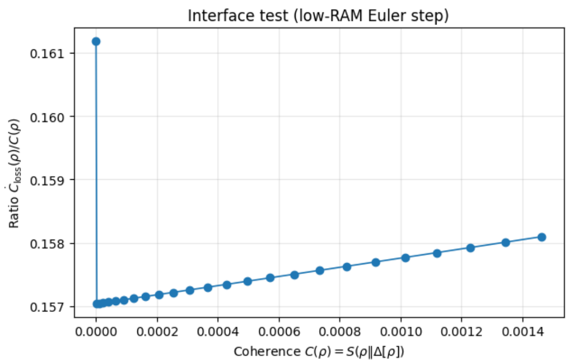
for the $\{\pm\omega\}$ truncation, and a slightly larger plateau

$$r(\rho) \approx 0.160\text{–}0.161$$

when an additional $\omega = 0$ channel with $\gamma(0) = 0.2$ is included.



(a) Linearity check: $\dot{C}_{\text{loss}}(\rho)$ versus $C(\rho)$.



(b) Ratio $r(\rho) = \dot{C}_{\text{loss}}(\rho)/C(\rho)$ versus $C(\rho)$.

Figure 4: State-side interface micro-test (Euler step with PSD projection). Ultra-small $C(\rho)$ points can be affected by eigenvalue clipping; the plateau region is stable.

$\omega = 0$-only regime: linear scaling in $\gamma(0)$. To isolate the commutator-squared mechanism of the negative lemma, we ran an $\omega = 0$-only truncation and chose (m, n) to maximize $|s_m - s_n|\sqrt{p_m p_n}$ where $s_k := kSk$ in the energy basis. With $\beta = 1$ and numerical prescription $\gamma(0) = \gamma_0/\beta$, the measured plateau ratio $r(\alpha = 0.15)$ scales linearly with $\gamma(0)$:

γ_0	$\gamma(0) = \gamma_0/\beta$	$r(\alpha = 0.15)$	$r/\gamma(0)$
0.05	0.05	0.0101957	0.2039
0.10	0.10	0.0203914	0.2039
0.40	0.40	0.0815656	0.2039

Table 4: TFIM $\omega = 0$-only micro-test ($N = 6$, $J = 1$, $h = 1.5$, $\beta = 1$). The empirical ratio $r = \dot{C}_{\text{loss}}/C$ scales linearly in $\gamma(0) = \gamma_0/\beta$.

Fixed-pair temperature scaling: $r(\beta) \propto 1/\beta$ at fixed γ_0. To avoid conflating temperature dependence with changes in the coherence sector, we fixed $(m, n) = (0, 1)$ and kept $\gamma_0 = 0.10$ fixed (so $\gamma(0) = \gamma_0/\beta$). The measured plateau satisfies $r(\beta) \approx k/\beta$:

β	$\gamma(0) = \gamma_0/\beta$ ($\gamma_0 = 0.10$)	$r(\alpha = 0.15)$	$\beta\, r(\alpha = 0.15)$
0.5	0.20	0.004317824675	0.002158912338
1.0	0.10	0.002158560224	0.002158560224
2.0	0.05	0.001078731632	0.002157463263
4.0	0.025	0.000538771757	0.002155087026

Table 5: Fixed-pair $\omega = 0$-only temperature sweep ($N = 6$, $J = 1$, $h = 1.5$, $(m, n) = (0, 1)$, $\gamma_0 = 0.10$). The near-constancy of βr confirms $r(\beta) \propto 1/\beta$ at fixed γ_0.

7 State–observable interface (explicit hypothesis)

The maintenance inequality bounds power in terms of the state functional $\dot{C}_{\text{loss}}(\rho)$, whereas $\kappa(\epsilon)$ is defined via a supremum over observables in $\mathcal{A}_\epsilon$. To avoid a quantifier mismatch, we state the necessary bridge as an explicit hypothesis.

Definition 2 (Weak-coherence regime on a family). *A family of states $\mathcal{F}_\epsilon$ lies in the weak-coherence regime if there exists $\delta_\epsilon > 0$ such that $C(\rho) \le \delta_\epsilon$ for all $\rho \in \mathcal{F}_\epsilon$.*

Definition 3 (Population non-extremality on a family). *A family $\mathcal{F}_\epsilon$ is population-nonextremal if there exists $p_{\min} > 0$ such that for all $\rho \in \mathcal{F}_\epsilon$,*

$$\Delta[\rho] \;\ge\; p_{\min}\mathbf{1}.$$

Remark 4. *This assumption rules out diagonals arbitrarily close to the boundary of the simplex, ensuring uniform local expansions of relative entropy around $\Delta[\rho]$ and preventing degenerate small-denominator effects in weak-coherence estimates.*

Definition 4 (Effective coherence sector). *Fix a set Ω of Bohr frequencies. We say ρ has coherence supported in Ω if all its off-diagonal matrix elements in the energy basis lie in frequency blocks $\omega \in \Omega$.*

Definition 5 (Sector rate lower bound). *For a Davies generator, let $\Gamma(\omega)$ denote the Kossakowski rate matrix in the Bohr block ω. In the single-channel case, $\Gamma(\omega)$ is a scalar and we identify $\lambda_{\min}(\Gamma(\omega)) = \Gamma(\omega)$. Define*

$$\underline{\Gamma}_\Omega := \min_{\omega\in\Omega} \lambda_{\min}(\Gamma(\omega)).$$

Hypothesis 1 (State–observable interface (lower proportionality on a family)). *Fix ϵ and a target family $\mathcal{F}_\epsilon$. Assume:*

1. *$\mathcal{F}_\epsilon$ lies in a weak-coherence regime;*
2. *$\mathcal{F}_\epsilon$ is population-nonextremal;*
3. *all $\rho \in \mathcal{F}_\epsilon$ have coherence supported in a fixed sector set Ω;*
4. *the Davies generator is validated in the regime of interest.*

Then there exists $\eta_\epsilon \to 0$ as the coherence scale $\delta_\epsilon \to 0$ such that for all $\rho \in \mathcal{F}_\epsilon$,

$$\dot{C}_{\text{loss}}(\rho) \;\ge\; (1-\eta_\epsilon)\,\underline{\Gamma}_\Omega\, C(\rho).$$

Remark 5. *This interface is falsifiable: one can numerically test whether $\dot{C}_{\text{loss}}(\rho)/C(\rho)$ is bounded below on the specified family.*

8 Operational consequence (conditional, well-typed)

Combining Theorem 1 with Hypothesis 1 yields, for $\rho \in \mathcal{F}_\epsilon$,

$$P_{\text{extra}}(\rho) \;\geq\; k_{\text{B}}T\,\dot{C}_{\text{loss}}(\rho) \;\geq\; k_{\text{B}}T\,(1-\eta_\epsilon)\,\underline{\Gamma}_\Omega\,C(\rho).$$

In particular, if $C(\rho) \geq C_0$ on $\mathcal{F}_\epsilon$, then

$$P_{\text{extra}}(\rho) \;\geq\; k_{\text{B}}T\,(1-\eta_\epsilon)\,\underline{\Gamma}_\Omega\,C_0.$$

9 Conclusion

We separated a proven finite-dimensional thermodynamic maintenance bound from conditional dynamical interface claims. In Davies dynamics, an exact $\omega = 0$ identity yields operational witness-based lower bounds. In TFIM witness computations, $R_{\text{opt}}(\epsilon)$ is robust and non-negligible at fixed separations, while decreasing as ϵ scales with N over the tested sizes.

We emphasize that the TFIM numerics ($N \leq 12$) are presented as finite-size witness diagnostics, not as definitive thermodynamic-limit scaling evidence.

A Numerical stability checks: time step and PSD clipping

We estimate $\dot{C}_{\text{loss}}(\rho)$ via a forward Euler step $\rho_t \approx \Pi(\rho + t\mathcal{L}(\rho))$ with PSD projection Π (eigenvalue clipping) and a finite-difference quotient.

dt	r (last sample)
$1e-4$	0.1613363453
$5e-5$	0.1613377388
$1e-5$	0.1613385899

Table 6: Sensitivity of the plateau ratio $r = \dot{C}_{\text{loss}}/C$ to the Euler step size dt (interface micro-test).

`psd_clip`	r (last sample)
$1e-12$	0.1613377388
$1e-14$	0.1613377388
$1e-16$	0.1613377388

Table 7: Sensitivity of the plateau ratio $r = \dot{C}_{\text{loss}}/C$ to the PSD projection threshold (interface micro-test).

B Imported maintenance inequality: precise pointer map

The main maintenance inequality used in this manuscript is

$$P_{\text{extra}}(\rho;\mathcal{L},T) \;\geq\; k_{\text{B}}T\,\dot{C}_{\text{loss}}(\rho), \qquad \dot{C}_{\text{loss}}(\rho) := -\frac{d}{dt}S(\rho_t\|\Delta[\rho_t])\Big|_{t=0}, \quad \rho_t = e^{t\mathcal{L}}(\rho).$$

A complete finite-dimensional proof is given in the published preprint [2].

C Reproducibility: Colab-friendly scripts (included inline)

This appendix includes scripts used to generate the state-side interface micro-tests (including figures) and the $\omega = 0$ fixed-pair β sweep.

C.1 Low-RAM interface micro-test script (Davies blocks; optional $\omega = 0$; pair selection)

```
# file: test_interface_hypothesis_tfim_lowram.py
import argparse
import numpy as np
import scipy.linalg as la
import matplotlib.pyplot as plt


def parse_args():
    ap = argparse.ArgumentParser()
    ap.add_argument("--N", type=int, default=6)
    ap.add_argument("--J", type=float, default=1.0)
    ap.add_argument("--h", type=float, default=1.5)
    ap.add_argument("--beta", type=float, default=1.0)

    ap.add_argument("--Sop", type=str, default="Z", choices=["X", "Y", "Z"])
    ap.add_argument("--center", action="store_true")
    ap.add_argument("--j0", type=int, default=None)

    ap.add_argument("--dt", type=float, default=5e-5)
    ap.add_argument("--nsamples", type=int, default=25)
    ap.add_argument("--alpha_max", type=float, default=0.15)

    ap.add_argument("--bin_tol", type=float, default=1e-10)
    ap.add_argument("--gamma0", type=float, default=0.2)
    ap.add_argument("--s", type=float, default=1.0)

    ap.add_argument("--use_omega0", action="store_true")
    ap.add_argument("--only_omega0", action="store_true")
    ap.add_argument("--only_pm_omega", action="store_true")

    ap.add_argument("--pair_mode", type=str, default="pmomega", choices=["pmomega", "
        omega0"])
    ap.add_argument("--psd_clip", type=float, default=1e-14)
    ap.add_argument("--seed", type=int, default=0)

    ap.add_argument("--save_prefix", type=str, default="interface")
    ap.add_argument("--no_plots", action="store_true")

    args, unknown = ap.parse_known_args()
    if unknown:
        print(f"Ignoring unknown args: {unknown}")

    if args.only_omega0 and args.only_pm_omega:
        raise ValueError("Choose at most one of --only_omega0 or --only_pm_omega.")
    if args.only_omega0 and (not args.use_omega0):
        raise ValueError("--only_omega0 requires --use_omega0.")

    if not args.center and args.j0 is None:
        args.center = True
```

```
    return args


def paulis():
    I = np.array([[1, 0], [0, 1]], dtype=complex)
    X = np.array([[0, 1], [1, 0]], dtype=complex)
    Y = np.array([[0, -1j], [1j, 0]], dtype=complex)
    Z = np.array([[1, 0], [0, -1]], dtype=complex)
    return I, X, Y, Z


def kron_all(ops):
    out = ops[0]
    for op in ops[1:]:
        out = np.kron(out, op)
    return out


def local_pauli(N, site, which):
    I, X, Y, Z = paulis()
    P = {"I": I, "X": X, "Y": Y, "Z": Z}[which.upper()]
    ops = [I] * N
    ops[site] = P
    return kron_all(ops)


def two_site_xx(N, i):
    I, X, Y, Z = paulis()
    ops = [I] * N
    ops[i] = X
    ops[i + 1] = X
    return kron_all(ops)


def tfim_hamiltonian(N, J, h):
    dim = 2**N
    H = np.zeros((dim, dim), dtype=complex)
    for i in range(N - 1):
        H += -J * two_site_xx(N, i)
    for i in range(N):
        H += -h * local_pauli(N, i, "Z")
    return (H + H.conj().T) / 2


def thermal_probs(E, beta):
    Emin = np.min(E.real)
    w = np.exp(-beta * (E.real - Emin))
    return w / np.sum(w)


def gamma_model(omega, beta, gamma0=0.2, s=1.0):
    if omega > 0:
        denom = 1.0 - np.exp(-beta * omega)
        denom = denom if denom > 1e-15 else 1e-15
        return float(gamma0 * (omega**s) / denom)
    if omega < 0:
        op = abs(omega)
        return float(np.exp(-beta * op) * gamma_model(op, beta, gamma0=gamma0, s=s))
```

```
    return 0.0


def project_to_density(rho, eps=1e-14):
    rho = 0.5 * (rho + rho.conj().T)
    evals, evecs = la.eigh(rho)
    evals = np.maximum(evals.real, eps)
    rho_psd = (evecs * evals) @ evecs.conj().T
    rho_psd = 0.5 * (rho_psd + rho_psd.conj().T)
    rho_psd = rho_psd / np.trace(rho_psd)
    return rho_psd


def rel_entropy(rho, sigma):
    rho = 0.5 * (rho + rho.conj().T)
    sigma = 0.5 * (sigma + sigma.conj().T)
    log_rho = la.logm(rho)
    log_sig = la.logm(sigma)
    return float(np.trace(rho @ (log_rho - log_sig)).real)


def lindblad_L_of_rho(rho, L_ops, rates):
    out = np.zeros_like(rho)
    for Lk, rk in zip(L_ops, rates):
        if rk == 0:
            continue
        LdL = Lk.conj().T @ Lk
        out += rk * (Lk @ rho @ Lk.conj().T - 0.5 * (LdL @ rho + rho @ LdL))
    return out


def build_Somega_bin(S_E, E, omega_target, bin_tol):
    d = S_E.shape[0]
    Sbin = np.zeros((d, d), dtype=complex)
    for m in range(d):
        for n in range(d):
            omega = float((E[m] - E[n]).real)
            if abs(omega - omega_target) <= bin_tol:
                Sbin[m, n] = S_E[m, n]
    return Sbin


def choose_pair_pmomega(S_E, p, E, bin_tol):
    d = S_E.shape[0]
    best = (-1.0, None)
    for m in range(d):
        for n in range(d):
            if m == n:
                continue
            omega = float((E[m] - E[n]).real)
            if omega <= bin_tol:
                continue
            score = float(abs(S_E[m, n]) * np.sqrt(p[m] * p[n]))
            if score > best[0]:
                best = (score, (m, n, omega))
    if best[1] is None:
        raise RuntimeError("Could not find a positive-frequency pair.")
    return best[1]
```

```
def choose_pair_omega0(S_E, p, E, bin_tol):
    d = S_E.shape[0]
    sdiag = np.diag(S_E).real
    best = (-1.0, None)
    for m in range(d):
        for n in range(d):
            if m == n:
                continue
            omega = float((E[m] - E[n]).real)
            if abs(omega) <= bin_tol:
                continue
            score = float(abs(sdiag[m] - sdiag[n]) * np.sqrt(p[m] * p[n]))
            if score > best[0]:
                best = (score, (m, n, abs(omega)))
    if best[1] is None:
        raise RuntimeError("Could not find a pair for omega0 mode.")
    return best[1]


def make_rho_E(p, m, n, alpha):
    rho_E = np.diag(p.astype(complex))
    amp = alpha * np.sqrt(p[m] * p[n])
    rho_E[m, n] += amp
    rho_E[n, m] += amp
    return rho_E


def main():
    args = parse_args()
    np.random.seed(args.seed)

    N, J, h, beta = args.N, args.J, args.h, args.beta
    j0 = args.j0 if args.j0 is not None else (N // 2 if args.center else 0)

    H = tfim_hamiltonian(N, J, h)
    E, V = la.eigh(H)
    Vdag = V.conj().T
    p = thermal_probs(E, beta)

    S = local_pauli(N, j0, args.Sop)
    S_E = Vdag @ S @ V

    if args.pair_mode == "pmomega":
        m, n, omega = choose_pair_pmomega(S_E, p, E, args.bin_tol)
        print("Pair mode: pmomega")
    else:
        m, n, omega = choose_pair_omega0(S_E, p, E, args.bin_tol)
        print("Pair mode: omega0")

    print(f"Selected pair (m,n)=({m},{n}), omega≈{omega:.12g}")

    Somega_E = build_Somega_bin(S_E, E, omega, args.bin_tol)
    Sminus_E = Somega_E.conj().T
    Somega = V @ Somega_E @ Vdag
    Sminus = V @ Sminus_E @ Vdag
```

```
    g_plus = gamma_model(omega, beta, gamma0=args.gamma0, s=args.s)
    g_minus = gamma_model(-omega, beta, gamma0=args.gamma0, s=args.s)

    omega0_ops, omega0_rates = [], []
    pm_ops, pm_rates = [Somega, Sminus], [g_plus, g_minus]

    if args.use_omega0:
        S0_E = np.diag(np.diag(S_E))
        S0 = V @ S0_E @ Vdag
        gamma0_val = float(args.gamma0 / beta)
        omega0_ops, omega0_rates = [S0], [gamma0_val]
        print(f"Included omega=0 channel with gamma(0)=gamma0/beta={gamma0_val:.6g}")

    if args.only_omega0:
        L_ops, rates = omega0_ops, omega0_rates
        print("Mode: ONLY omega=0")
    elif args.only_pm_omega:
        L_ops, rates = pm_ops, pm_rates
        print("Mode: ONLY ±omega")
    else:
        L_ops, rates = omega0_ops + pm_ops, omega0_rates + pm_rates
        print("Mode: omega=0 (if enabled) PLUS ±omega")

    alphas = np.linspace(1e-4, args.alpha_max, args.nsamples)
    dt = args.dt

    Cs, dCs, ratios = [], [], []
    for alpha in alphas:
        rho_E = make_rho_E(p, m, n, alpha)
        rho = project_to_density(V @ rho_E @ Vdag, eps=args.psd_clip)

        rho_E_chk = Vdag @ rho @ V
        Delta_rho = project_to_density(V @ np.diag(np.diag(rho_E_chk)) @ Vdag, eps=args
            .psd_clip)
        C = rel_entropy(rho, Delta_rho)

        Lrho = lindblad_L_of_rho(rho, L_ops, rates)
        rho_t = project_to_density(rho + dt * Lrho, eps=args.psd_clip)

        rho_t_E = Vdag @ rho_t @ V
        Delta_rho_t = project_to_density(V @ np.diag(np.diag(rho_t_E)) @ Vdag, eps=args
            .psd_clip)
        C_t = rel_entropy(rho_t, Delta_rho_t)

        dotC_loss = -(C_t - C) / dt

        Cs.append(C)
        dCs.append(dotC_loss)
        ratios.append(dotC_loss / C if C > 0 else np.nan)

    Cs = np.array(Cs, dtype=float)
    dCs = np.array(dCs, dtype=float)
    ratios = np.array(ratios, dtype=float)

    alpha_last = float(alphas[-1])
    r_last = float(ratios[-1])
    C_last = float(Cs[-1])
```

```
    print(f"RESULT alpha_last={alpha_last:.12g} C_last={C_last:.12g} r_last={r_last
        :.12g}")

    if not args.no_plots:
        prefix = args.save_prefix

        plt.figure(figsize=(6.5, 4.2))
        plt.plot(Cs, ratios, marker="o", linewidth=1.2)
        plt.grid(True, alpha=0.3)
        plt.xlabel("Coherence $C(\\rho)=S(\\rho\\|\\Delta[\\rho])$")
        plt.ylabel("Ratio $\\dot C_{\\mathrm{loss}}(\\rho)/C(\\rho)$")
        plt.title("Interface test (low-RAM Euler step)")
        plt.tight_layout()
        plt.savefig(f"{prefix}_ratio_vs_C.png", dpi=200)

        plt.figure(figsize=(6.5, 4.2))
        plt.plot(Cs, dCs, marker="o", linewidth=1.2)
        plt.grid(True, alpha=0.3)
        plt.xlabel("$C(\\rho)$")
        plt.ylabel("$\\dot C_{\\mathrm{loss}}(\\rho)$")
        plt.title("Interface test: linearity check (low-RAM Euler)")
        plt.tight_layout()
        plt.savefig(f"{prefix}_dotC_vs_C.png", dpi=200)

        print(f"Saved -> {prefix}_ratio_vs_C.png")
        print(f"Saved -> {prefix}_dotC_vs_C.png")


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

C.2 $\omega = 0$-only fixed-pair sweep script (minimal, for the β table)

```
# file: tfim_omega0_fixedpair.py
import argparse
import numpy as np
import scipy.linalg as la


def parse_args():
    ap = argparse.ArgumentParser()
    ap.add_argument("--N", type=int, default=6)
    ap.add_argument("--J", type=float, default=1.0)
    ap.add_argument("--h", type=float, default=1.5)
    ap.add_argument("--beta", type=float, default=1.0)
    ap.add_argument("--Sop", type=str, default="Z", choices=["X", "Y", "Z"])
    ap.add_argument("--center", action="store_true")
    ap.add_argument("--j0", type=int, default=None)

    ap.add_argument("--m", type=int, required=True)
    ap.add_argument("--n", type=int, required=True)

    ap.add_argument("--gamma0", type=float, default=0.10, help="Implements gamma(0)=
        gamma0/beta")
    ap.add_argument("--dt", type=float, default=5e-5)
    ap.add_argument("--alpha_max", type=float, default=0.15)
    ap.add_argument("--nsamples", type=int, default=25)
```

```python
    ap.add_argument("--psd_clip", type=float, default=1e-14)
    ap.add_argument("--seed", type=int, default=0)
    return ap.parse_args()


def paulis():
    I = np.array([[1, 0], [0, 1]], dtype=complex)
    X = np.array([[0, 1], [1, 0]], dtype=complex)
    Y = np.array([[0, -1j], [1j, 0]], dtype=complex)
    Z = np.array([[1, 0], [0, -1]], dtype=complex)
    return I, X, Y, Z


def kron_all(ops):
    out = ops[0]
    for op in ops[1:]:
        out = np.kron(out, op)
    return out


def local_pauli(N, site, which):
    I, X, Y, Z = paulis()
    P = {"X": X, "Y": Y, "Z": Z}[which.upper()]
    ops = [I] * N
    ops[site] = P
    return kron_all(ops)


def two_site_xx(N, i):
    I, X, Y, Z = paulis()
    ops = [I] * N
    ops[i] = X
    ops[i + 1] = X
    return kron_all(ops)


def tfim_hamiltonian(N, J, h):
    dim = 2**N
    H = np.zeros((dim, dim), dtype=complex)
    for i in range(N - 1):
        H += -J * two_site_xx(N, i)
    for i in range(N):
        H += -h * local_pauli(N, i, "Z")
    return (H + H.conj().T) / 2


def thermal_probs(E, beta):
    Emin = np.min(E.real)
    w = np.exp(-beta * (E.real - Emin))
    return w / np.sum(w)


def project_to_density(rho, eps=1e-14):
    rho = 0.5 * (rho + rho.conj().T)
    evals, evecs = la.eigh(rho)
    evals = np.maximum(evals.real, eps)
    rho_psd = (evecs * evals) @ evecs.conj().T
    rho_psd = 0.5 * (rho_psd + rho_psd.conj().T)
```

```python
    rho_psd = rho_psd / np.trace(rho_psd)
    return rho_psd


def rel_entropy(rho, sigma):
    rho = 0.5 * (rho + rho.conj().T)
    sigma = 0.5 * (sigma + sigma.conj().T)
    return float(np.trace(rho @ (la.logm(rho) - la.logm(sigma))).real)


def lindblad_omega0_L(rho, S0, gamma0_val):
    S0sq = S0 @ S0
    return gamma0_val * (S0 @ rho @ S0 - 0.5 * (S0sq @ rho + rho @ S0sq))


def main():
    args = parse_args()
    np.random.seed(args.seed)

    N, J, h, beta = args.N, args.J, args.h, args.beta
    j0 = args.j0 if args.j0 is not None else (N // 2 if args.center else 0)

    H = tfim_hamiltonian(N, J, h)
    E, V = la.eigh(H)
    Vdag = V.conj().T
    p = thermal_probs(E, beta)

    S = local_pauli(N, j0, args.Sop)
    S_E = Vdag @ S @ V

    S0_E = np.diag(np.diag(S_E))
    S0 = V @ S0_E @ Vdag

    m, n = args.m, args.n
    omega = abs(float((E[m] - E[n]).real))
    smsn = abs((S_E[m, m] - S_E[n, n]).real)
    gamma0_val = float(args.gamma0 / beta)

    print(f"Fixed pair (m,n)=({m},{n}), omega≈{omega:.12g}")
    print(f"beta={beta}, gamma0={args.gamma0}, gamma(0)=gamma0/beta={gamma0_val}")
    print(f"p_m={p[m]:.6g}, p_n={p[n]:.6g}, |s_m-s_n|={smsn:.6g}")

    alphas = np.linspace(1e-4, args.alpha_max, args.nsamples)
    dt = args.dt

    diag_E = np.diag(p.astype(complex))

    print("\nalpha C(rho) dotC_loss dotC_loss/C")
    last_ratio = None
    for alpha in alphas:
        rho_E = diag_E.copy()
        amp = alpha * np.sqrt(p[m] * p[n])
        rho_E[m, n] += amp
        rho_E[n, m] += amp

        rho = project_to_density(V @ rho_E @ Vdag, eps=args.psd_clip)

        rho_E_chk = Vdag @ rho @ V
```

```python
        Delta_rho_E = np.diag(np.diag(rho_E_chk))
        Delta_rho = project_to_density(V @ Delta_rho_E @ Vdag, eps=args.psd_clip)

        C = rel_entropy(rho, Delta_rho)

        rho_t = project_to_density(
            rho + dt * lindblad_omega0_L(rho, S0, gamma0_val),
            eps=args.psd_clip
        )

        rho_t_E = Vdag @ rho_t @ V
        Delta_rho_t_E = np.diag(np.diag(rho_t_E))
        Delta_rho_t = project_to_density(V @ Delta_rho_t_E @ Vdag, eps=args.psd_clip)

        C_t = rel_entropy(rho_t, Delta_rho_t)
        dotC_loss = -(C_t - C) / dt
        ratio = dotC_loss / C if C > 0 else float("nan")
        last_ratio = ratio
        print(f"{alpha: .6f} {C: .6e} {dotC_loss: .6e} {ratio: .6g}")

    print(f"\nRESULT r(alpha={args.alpha_max}) = {last_ratio}")


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

References

[1] L. Eriksson, *Clustering, Recovery, and Locality in Algebraic Quantum Field Theory: Quantitative Bounds via Split Inclusions and Modular Theory*, ai.viXra:2512.0060 (2025). https://ai.vixra.org/abs/2512.0060

[2] L. Eriksson, *The Conditional Maintenance Work Theorem: Operational Power Lower Bounds from Energy Pinching and a Split-Inclusion Blueprint for Type III AQFT*, ai.viXra:2512.0061 (2025). https://ai.vixra.org/abs/2512.0061

[3] L. Eriksson, *The Heisenberg Cut as a Resource Boundary: An Operational Outlook from Coherence Maintenance Costs*, ai.viXra:2512.0064 (2025). https://ai.vixra.org/abs/2512.0064

[4] E. B. Davies, *Markovian master equations*, Commun. Math. Phys. **39** (1974), 91–110.

[5] H. Spohn, *Entropy production for quantum dynamical semigroups*, J. Math. Phys. **19** (1978), 1227–1230.

[6] H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems*, Oxford University Press (2002).