Computing quantum magic of state vectors

Piotr Sierant,[1] Jofre Vallès-Muns,[1] and Artur Garcia-Saez[1]

[1]*Barcelona Supercomputing Center, Barcelona 08034, Spain*

(Dated: January 13, 2026)

Non-stabilizerness, also known as "magic," quantifies how far a quantum state departs from the stabilizer set. It is a central resource behind quantum advantage and a useful probe of the complexity of many-body quantum states. Yet standard magic quantifiers, such as the stabilizer Rényi entropy (SRE) for qubits and the mana for qutrits, are costly to evaluate numerically, with the computational complexity growing rapidly with the number N of qudits. Here we introduce efficient, numerically exact algorithms that exploit the fast Hadamard transform to compute the SRE for qubits ($d = 2$) and the mana for qutrits ($d = 3$) for pure states given as state vectors. Our methods reduce the runtime to $O(Nd^{2N})$, an *exponential* improvement over the naive $O(d^{3N})$ scaling, while exposing substantial parallelism and enabling GPU acceleration. We further show how to combine the fast Hadamard transform with Monte Carlo sampling to estimate the SRE of state vectors, and we extend the approach to compute the mana of mixed states. All algorithms are implemented in the open-source Julia package HadaMAG.jl, which provides a high-performance, GPU-enabled toolbox for computing SRE and mana. The package, together with the methods developed in this work, offers a practical route to large-scale numerical studies of magic in quantum many-body systems.

I. INTRODUCTION

A quantum state of N qudits is represented by a vector in a Hilbert space of dimension $\mathcal{N} = d^N$. This *exponential* growth suggests that sufficiently large quantum devices may become intractable to simulate classically, enabling quantum computational advantage [1, 2]. At the same time, many physically relevant states possess structure that can be leveraged for compact classical descriptions. For example, weakly entangled states admit efficient tensor-network representations [3–7], which have been central to our understanding of equilibrium many-body physics. In particular, ground states of one-dimensional local gapped Hamiltonians are well captured by matrix product states [8–10]. Beyond tensor networks, other families of many-body states are also classically tractable, including fermionic Gaussian states [11–13] and stabilizer states [14, 15]. Although such states can exhibit extensive entanglement, they remain efficiently simulable with resources that scale *polynomially* in system size N. This motivates quantitative notions of fermionic non-Gaussianity [16] and non-stabilizerness [17], which characterize how far a state lies from efficiently simulable classes such as Gaussian or stabilizer states.

Measures of non-stabilizerness are monotonic under stabilizer protocols [18–20], which include Clifford operations [21], but can increase under non-stabilizer gates such as the T gate [22]. Together with Clifford gates, the T gate completes a universal gate set [23]. Several measures of non-stabilizerness have been proposed, including stabilizer rank [24, 25], stabilizer fidelity [26], and robustness of magic [27–29]. However, evaluating these measures typically requires costly optimization procedures that become prohibitive already for systems with fewer than $N = 10$ qubits [30].

In contrast, the recently introduced *stabilizer Rényi entropy* (SRE) [31] avoids optimization while retaining the key properties expected of a qubit ($d = 2$) non-stabilizerness measure [32]. Its introduction has sparked extensive efforts to characterize non-stabilizerness across a broad range of many-body settings, including quantum phases and phase transitions [33–43], nonequilibrium dynamics [44–59], and experimental measurements [60–62].

For qutrits ($d = 3$), the structure of the Pauli group [63] enables the definition of a discrete Wigner

function [64, 65], whose negativity yields a non-stabilizerness measure known as *mana* [18, 66, 67]. Like the SRE, mana avoids costly minimization, which has enabled studies in many-body ground states [68, 69] as well as in out-of-equilibrium settings [46, 55, 70–76]. More broadly, investigations of non-stabilizerness extend beyond quantum computation and condensed matter physics, with applications in nuclear physics [77, 78], particle physics [79, 80], quantum electrodynamics [81], quantum chemistry [82], and the AdS/CFT correspondence [71, 83].

The ability to compute measures of non-stabilizerness is central to understanding magic-state resources in many-body systems. Although the SRE and mana avoid explicit minimization, their evaluation remains challenging because it requires computing $\mathcal{N}^2 = d^{2N}$ expectation values of highly nonlocal observables. For quantum states given as state vectors, a direct evaluation of these d^{2N} quantities [84] is feasible on present-day hardware only up to $N \approx 12$–14 qubits [40, 78, 85]. For weakly entangled states, which are well approximated by tensor networks, several approaches have been developed to compute the SRE [33, 34, 86–88] and mana [69], enabling studies of systems comprising hundreds of qudits.

However, many physically relevant quantum states are not weakly entangled. Examples include nonequilibrium dynamics [89, 90], highly excited eigenstates of many-body Hamiltonians [91, 92], and systems with long-range interactions [93, 94]. In such settings, numerical methods can still produce state vectors for systems with more than $N = 30$ qubits [95], far beyond the regime where direct SRE evaluation is practical [84] and, for highly entangled states, beyond the reach of tensor-network-based methods of non-stabilizerness calculation.

This work fills this gap by introducing efficient algorithms to compute the SRE and mana for quantum states represented as state vectors. Leveraging the fast Hadamard transform [96], we develop numerically exact procedures for evaluating SRE and mana that achieve an exponential speedup in the system size N compared to direct numerical methods [84], without additional memory overhead. This enables numerically exact computations of these non-stabilizerness measures for systems with up to $N = 25$ qubits and $N = 15$ qutrits. We further use the Hadamard transform to devise a sampling-based estimator for the SRE, extending the range of system sizes for which SRE can be computed from state vectors. Finally, we introduce a fast-Hadamard-transform-based method to compute the mana of mixed states.

This manuscript is structured as follows. In Sec. II we introduce the non-stabilizerness measures of interest: the SRE and mana. Sec. III contains our main contributions and presents (i) in Sec. III A 2, a numerically exact algorithm for SRE; (ii) in Sec. III B, a sampling algorithm for SRE; (iii) in Sec. III C, a numerically exact algorithm for the mana of pure states; and (iv) in Sec. III D, an algorithm for the mana of mixed states. All these algorithms are implemented in the open-source package HadaMAG.jl. In Sec. IV we demonstrate the use of this package and benchmark our methods on states generated by random quantum circuits. Concluding remarks and an outlook are given in Sec. V.

II. NON-STABILIZERNESS MEASURES

We consider a system of N qubits ($d = 2$), and denote by I, X, Y, and Z the identity and the Pauli X, Y, and Z operators. Pauli strings are operators of the form $P = P_1 \otimes P_2 \otimes \cdots \otimes P_N$, where each $P_j \in \{I, X, Y, Z\}$. The set of all N-qubit Pauli strings (modulo phases $\pm 1, \pm i$) will be denoted by $\mathcal{P}_N$. An element $P \in \mathcal{P}_N$ can be parametrized by two binary strings $\mathbf{a} = (a_1, \ldots, a_N) \in \mathbb{Z}_2^N$ and

$\mathbf{b} = (b_1, \ldots, b_N) \in \mathbb{Z}_2^N$, with $a_j, b_j \in \{0, 1\}$, as

$$P_{\mathbf{a},\mathbf{b}} = \prod_{j=1}^N X_j^{a_j} \prod_{j=1}^N Z_j^{b_j}, \tag{1}$$

up to an overall phase. The Clifford group $\mathcal{C}_{N,2}$ is the subgroup of the unitary group that maps Pauli strings to Pauli strings (up to phases $\pm 1, \pm i$). Stabilizer states, denoted by $|\sigma\rangle$, are the states obtained by the action of $\mathcal{C}_{N,2}$ on the initial state $|0\rangle^{\otimes N}$.

The stabilizer Rényi entropy (SRE), introduced in Ref. [31], is defined for a pure state $|\psi\rangle$ of N qubits as

$$M_q(|\psi\rangle) = \frac{1}{1-q} \log_2 \left[\sum_{P \in \mathcal{P}_N} \frac{\langle\psi|P|\psi\rangle^{2q}}{2^N} \right] = \frac{1}{1-q} \log_2 \left[\sum_{\mathbf{a},\mathbf{b}\in\mathbb{Z}_2^N} \frac{\langle\psi|P_{\mathbf{a},\mathbf{b}}|\psi\rangle^{2q}}{2^N} \right], \tag{2}$$

where $q > 0$ is the Rényi index, and the sum extends over all 4^N Pauli strings $P_{\mathbf{a},\mathbf{b}}$. The quantity M_1 is defined via the limit $q \to 1$ in (2).

The SRE satisfies the key properties expected of a non-stabilizerness measure for many-body systems:

- faithfulness: $M_q(|\psi\rangle) \geq 0$, with equality if and only if $|\psi\rangle$ is a stabilizer state.

- Clifford invariance: $M_q(C|\psi\rangle) = Mq(|\psi\rangle)$ for any Clifford unitary $C \in \mathcal{C}N, d$, i.e., any unitary that maps Pauli strings P to Pauli strings $P' = \omega, CPC^\dagger$ up to a global phase $\omega \in \pm 1, \pm i$.

- additivity: $M_q(|\Phi\rangle \otimes |\psi\rangle) = M_q(|\Phi\rangle) + M_q(|\psi\rangle)$.

Clifford invariance ensures that the SRE remains unchanged under the action of Clifford gates. In the resource theory of non-stabilizerness, the convex hull $\mathsf{STAB} \equiv \{\sum_i p_i |\sigma_i\rangle\langle\sigma_i| : p_i > 0, , \sum_i p_i = 1\}$ of pure stabilizer states $|\sigma_i\rangle$ defines the set of free states. The free operations—referred to as stabilizer protocols—are completely positive trace-preserving (CPTP) maps that leave the set STAB invariant. These protocols consist of the following elementary operations: i) action of Clifford unitaries; (ii) partial trace; (iii) measurements in the computational basis; (iv) composition with auxiliary qubits in state $|0\rangle$; (v) operations (i)–(iv) conditioned on the measurement outcomes and classical randomness.

The SRE for every integer $q \geq 2$ has been shown to be a pure-state stabilizer monotone [97]. That is, for any stabilizer protocol $\mathcal{E}$ such that $\mathcal{E}(|\psi\rangle\langle\psi|) = |\phi\rangle\langle\phi|$, the SRE is non-increasing:

$$M_q(|\psi\rangle) \geq M_q(|\phi\rangle), \tag{3}$$

provided that $q \geq 2$ is an integer. In contrast, for $0 < q < 2$, M_q may increase under stabilizer protocols [20]. The SRE admits an operational interpretation in the context of quantum property testing [97] and is bounded from above by other non-stabilizerness measures such as the robustness of magic [31] and the stabilizer extent [25]. These alternative measures, however, involve minimization procedures that become computationally intractable beyond just a few qubits.

A. Mana for qutrits

For qutrits ($d = 3$) [98], we define shift and clock operators as

$$X = \sum_{k=0}^{d-1} |k+1\rangle\langle k| \quad \text{and} \quad Z = \sum_{k=0}^{d-1} \omega_d^k |k\rangle\langle k|, \tag{4}$$

where $\omega_d = e^{i2\pi/d}$ and the addition is defined modulo d. The generalized Pauli operators, also known as the Heisenberg-Weyl operators, are defined as

$$T_{ab} = \omega_d^{-2^{-1}ab} Z^a X^b \tag{5}$$

for $a, b \in \mathbb{Z}_d$ and with 2^{-1} the inverse element of 2 in $\mathbb{Z}_d$. For a system of N qutrits, the generalized Pauli strings read

$$T_{\mathbf{u}} = T_{u_1, u'_1} \otimes T_{u_2, u'_2} \otimes ... \otimes T_{u_N, u'_N}, \tag{6}$$

where $\mathbf{u} = (u_1, u'_1, \ldots, u_N, u'_N) \in \mathbb{Z}_d^{2N}$. The Clifford group $\mathcal{C}_{N,d}$ maps, up to a phase, generalized Pauli strings to generalized Pauli strings, and stabilizer states of qutrits are obtained by action of $\mathcal{C}_{N,d}$ on $|0\rangle^N$.

The phase-space point operators, defined in terms of the Pauli strings as

$$A_{\mathbf{0}} = \frac{1}{d^N} \sum_{\mathbf{u} \in \mathbb{Z}_d^{2N}} T_{\mathbf{u}}, \quad A_{\mathbf{u}} = T_{\mathbf{u}} A_{\mathbf{0}} T_{\mathbf{u}}^{\dagger}, \tag{7}$$

are orthogonal, i.e, $\text{tr}(A_{\mathbf{u}} A_{\mathbf{u}'}) = d^N \delta(\mathbf{u}, \mathbf{u}')$, and thus they provide an orthogonal basis for the density matrix ρ of a state as

$$\rho = \sum_{\mathbf{u} \in \mathbb{Z}_d^{2N}} W_\rho(\mathbf{u}) A_{\mathbf{u}}. \tag{8}$$

where $W_\rho(\mathbf{u}) = \frac{1}{d^N} \text{tr}(A_{\mathbf{u}}\rho)$ is the discrete Wigner function [64]. Finally, mana [18], defined in terms of the Wigner function as

$$\mathcal{M}(\rho) = \log_2 \left[\sum_{\mathbf{u} \in \mathbb{Z}_d^{2N}} |W_\rho(\mathbf{u})| \right], \tag{9}$$

due to the normalization condition $\sum_{\mathbf{u}} W_\rho(\mathbf{u}) = 1$, measures the negativity of the Wigner representation of ρ.

Mana is a faithful measure of non-stabilizerness for qutrit systems. Indeed, by the discrete Hudson theorem, a pure state $\rho = |\psi\rangle\langle\psi|$ has a nonnegative discrete Wigner function, $W_{|\psi\rangle\langle\psi|}(\mathbf{u}) \geq 0$, if and only if it is a stabilizer state [64]. Mana further satisfies the basic requirements of Clifford invariance and additivity, and it is a monotone not only for pure states [cf.(3)] but also for mixed states [18]. Notably, mana is the only known non-stabilizerness monotone whose definition avoids any minimization procedure for both pure and mixed states.

In the following, we will be considering calculation of mana for pure states, $\rho = |\psi\rangle\langle\psi|$, for which Eq. (9) reduces to

$$\mathcal{M}(|\psi\rangle) = \log_2\left[\sum_{A\in A_N}\frac{|\langle\psi|A|\psi\rangle|}{3^N}\right] = \log_2\left[\sum_{\mathbf{u}\in\mathbb{Z}_d^{2N}}\frac{|\langle\psi|A_\mathbf{u}|\psi\rangle|}{3^N}\right],\tag{10}$$

where we have introduced the set $A_N = \{A_\mathbf{u}\}_{\mathbf{u}\in\mathbb{Z}_d^{2N}}$ of d^{2N} phase-space point operators (7), to highlight the similarity between the definition of mana for pure states and the SRE (2).

III. ALGORITHMS FOR COMPUTATION OF SRE AND MANA

Equations (2) and (10) show that evaluating the SRE and the mana requires computing $\mathcal{N}^2 = d^{2N}$ expectation values of Pauli strings P and phase-space point operators A, respectively. While conceptually straightforward, this quickly becomes computationally prohibitive with the increase of the system size N. In this section, we present algorithms for the efficient evaluation of the SRE and mana for quantum states represented as state vectors.

A. Calculation of SRE for qubits

1. Naive SRE calculation

A direct numerical evaluation of the SRE in Eq. (2) requires computing

$$S_q = \sum_{\mathbf{a},\mathbf{b}\in\mathbb{Z}_2^N}\langle\psi|P_{\mathbf{a},\mathbf{b}}|\psi\rangle^{2q},\tag{11}$$

i.e., a sum over 4^N expectation values of Pauli strings $P_{\mathbf{a},\mathbf{b}}$ in the state $|\psi\rangle$. A straightforward approach is to generate the bit strings $\mathbf{a} = (a_j)$ and $\mathbf{b} = (b_j)$ in lexicographic order and, for each pair $(\mathbf{a}, \mathbf{b})$, compute $|\phi\rangle = P_{\mathbf{a},\mathbf{b}}|\psi\rangle$ and then the overlap $\langle\psi|\phi\rangle$. In practice, forming $P_{\mathbf{a},\mathbf{b}}|\psi\rangle$ may require applying up to $2N$ single-qubit operators X_j and Z_j, represented as sparse matrices. Generating $\mathbf{a}$ and $\mathbf{b}$ in Gray-code order [99], so that successive bit strings differ only by a single bit, reduces this cost: each step requires applying only a single X_j or Z_j operator. This leads to the following algorithm.

Algorithm 1. Naive calculation of SRE, $M_q(|\psi\rangle)$, via Gray-code sweep of Pauli strings.

Require: number of qubits N; Rényi index $q \neq 1$, state $|\psi\rangle$

$\quad S_q \leftarrow 0$, $\mathbf{a} \leftarrow (0,\ldots,0)$, $\mathbf{b} \leftarrow (0,\ldots,0)$,

$\quad |\phi\rangle \leftarrow |\psi\rangle$ $\qquad\qquad\qquad\qquad\triangleright$ initial Pauli string is the identity; at each step $|\phi\rangle = P|\psi\rangle$

$\quad$**for** $k = 0$ to $2^N - 1$ **do**

$\quad\quad$**for** $l = 0$ to $2^N - 1$ **do**

$\quad\quad\quad S_q \leftarrow S_q + \langle\psi|\phi\rangle^{2q}$

$\quad\quad\quad \mathbf{b}, t \leftarrow \text{NEXTGRAY}_2(\mathbf{b})$

$\quad\quad\quad |\phi\rangle \leftarrow Z_t|\phi\rangle$

```
        end for
        a, w ← NEXTGRAY₂(a)
        |φ⟩ ← X_w |φ⟩
    end for
    return M_q(|ψ⟩) = 1/(1−q) log₂(S_q/2^N).
```

Since two Pauli strings P and Q either commute ($PQ = QP$) or anticommute ($PQ = -QP$), the state $|\phi\rangle$ produced at each step of the algorithm is, up to an overall sign ± 1, equal to $|\phi\rangle = P_{\mathbf{a},\mathbf{b}} |\psi\rangle$. The above procedure requires evaluating 4^N overlaps $\langle\psi|\phi\rangle$, each at computational cost $O(2^N)$, as well as performing 4^N single-Pauli updates of the state, $|\phi\rangle \leftarrow Z_t |\phi\rangle$ or $|\phi\rangle \leftarrow X_w |\phi\rangle$, each also costing $O(2^N)$. Consequently, the overall time complexity of Algorithm 1 is $O(8^N)$, while the memory footprint is $O(2^N)$, set by storing the state vectors $|\psi\rangle$ and $|\phi\rangle$.

2. Efficient algorithm for exact SRE calculation

Each Pauli string $P_{\mathbf{a},\mathbf{b}}$ appearing in the SRE calculation can be written as a product $X_{\mathbf{a}} Z_{\mathbf{b}}$, where $X_{\mathbf{a}} = \prod_{j=1}^N X_j^{a_j}$ contains only the X_j operators and $Z_{\mathbf{b}} = \prod_{j=1}^N Z_j^{b_j}$ contains only the Z_j operators. A key ingredient behind the exponential speedup over Algorithm 1 is that, for a fixed X-string $X_{\mathbf{a}}$, the entire inner loop over all Z-strings $Z_{\mathbf{b}}$ can be replaced by a single fast Hadamard transform. In Algorithm 1, the inner Gray-code sweep updates $\mathbf{b} \in \mathbb{Z}_2^N$, applies $|\phi\rangle \leftarrow Z_t |\phi\rangle$, and accumulates the overlaps $\langle\psi|\phi\rangle^{2q} = \langle\psi|X_{\mathbf{a}} Z_{\mathbf{b}}|\psi\rangle^{2q}$. Equivalently, denoting $|\psi'\rangle = X_{\mathbf{a}} |\psi\rangle$, the inner loop computes $\chi_{\mathbf{b}} \equiv \langle\psi'| Z_{\mathbf{b}} |\psi\rangle$ for all $\mathbf{b} \in \mathbb{Z}_2^N$.

Expanding $|\psi'\rangle = \sum_{\mathbf{x}} \beta_{\mathbf{x}} |\mathbf{x}\rangle$ and $|\psi\rangle = \sum_{\mathbf{x}} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$ in the computational basis $\{|\mathbf{x}\rangle\}$ with $\mathbf{x} \in \mathbb{Z}_2^N$, and using $Z_{\mathbf{b}} |\mathbf{x}\rangle = (-1)^{\mathbf{b}\cdot\mathbf{x}} |\mathbf{x}\rangle$, we obtain the explicit expression

$$\chi_{\mathbf{b}} = \sum_{\mathbf{x}\in\mathbb{Z}_2^N} \beta_{\mathbf{x}}^* \alpha_{\mathbf{x}} (-1)^{\mathbf{b}\cdot\mathbf{x}}. \tag{12}$$

Thus, the length-2^N vector $(\chi_{\mathbf{b}})_{\mathbf{b}\in\mathbb{Z}_2^N}$ is the Walsh–Hadamard transform (i.e., the Fourier transform over $\mathbb{Z}_2^N$) of the vector with components $v_{\mathbf{x}} \equiv \beta_{\mathbf{x}}^* \alpha_{\mathbf{x}}$. In matrix form,

$$(\chi_{\mathbf{b}})_{\mathbf{b}\in\mathbb{Z}_2^N} = H_2^{\otimes N} (v_{\mathbf{x}})_{\mathbf{x}\in\mathbb{Z}_2^N}, \tag{13}$$

where $H_2^{\otimes N}$ is the N-fold tensor product of the unnormalized single-qubit Hadamard matrix $H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Computing $(\chi_{\mathbf{b}})_{\mathbf{b}\in\mathbb{Z}_2^N}$ by explicit matrix multiplication would require a computational cost $O(4^N)$, whereas the fast Hadamard transform evaluates Eq. (13) in $O(N2^N)$ time [96]. Since this computation must be repeated for all 2^N choices of $\mathbf{a}$, the resulting overall cost is $O(N4^N)$, yielding an exponential improvement over the $O(8^N)$ scaling of Algorithm 1. This leads to the following algorithm for the numerically exact calculation of the SRE.

Algorithm 2. Calculation of the SRE $M_q(|\psi\rangle)$ using the fast Hadamard transform.

Require: Number of qubits N; Rényi index $q \neq 1$; state $|\psi\rangle$.
    $S_q \leftarrow 0$, $\mathbf{a} \leftarrow (0,\ldots,0) \in \mathbb{Z}_2^N$, $|\psi'\rangle \leftarrow |\psi\rangle$ ▷ $|\psi'\rangle = \left(\prod_{j=1}^N X_j^{a_j}\right) |\psi\rangle$.
    for $k = 0$ to $2^N - 1$ **do**

Extract amplitudes $\beta_\mathbf{x}$ of $|\psi'\rangle = \sum_\mathbf{x} \beta_\mathbf{x} |\mathbf{x}\rangle$ and $\alpha_\mathbf{x}$ of $|\psi\rangle = \sum_\mathbf{x} \alpha_\mathbf{x} |\mathbf{x}\rangle$.
Form $v_\mathbf{x} \leftarrow \beta_\mathbf{x}^* \alpha_\mathbf{x}$ for all $\mathbf{x} \in \mathbb{Z}_2^N$.
$(\chi_\mathbf{b})_\mathbf{b} \leftarrow \text{FASTHADAMARDTRANSFORM}((v_\mathbf{x})_\mathbf{x})$.
$S_q \leftarrow S_q + \sum_{\mathbf{b} \in \mathbb{Z}_2^N} \chi_\mathbf{b}^{2q}$ ▷ Accumulates $\sum_\mathbf{b} \langle \psi' | Z_\mathbf{b} | \psi \rangle^{2q}$.
$\mathbf{a}, t \leftarrow \text{NEXTGRAY}_2(\mathbf{a})$ ▷ Advance Gray code for $\mathbf{a}$; t is the flipped bit.
$|\psi'\rangle \leftarrow X_t |\psi'\rangle$.
end for
return $M_q(|\psi\rangle) = \frac{1}{1-q} \log_2(S_q/2^N)$.

Algorithm 2 has time complexity $O(N4^N)$ and memory footprint $O(2^N)$, set by storing the state vectors $|\psi\rangle$, $|\psi'\rangle$, and a work array of length 2^N.

B.   Monte Carlo sampling algorithm for SRE

1.   Direct sampling of Pauli strings

Treating the Pauli strings $P \in \mathcal{P}_N$ as a basis of the operator space, any pure state $|\psi\rangle$ of N qubits with density matrix $\rho = |\psi\rangle\langle\psi|$ admits the expansion $\rho = 2^{-N} \sum_{P \in \mathcal{P}_N} \langle \psi | P | \psi \rangle\, P$. The normalization of the state, $\| |\psi\rangle \| = 1$, then implies that $\sum_{P \in \mathcal{P}_N} \langle \psi | P | \psi \rangle^2 / 2^N = 1$, which leads to a natural probability distribution over Pauli strings,

$$\pi(P) \; = \; \frac{\langle \psi | P | \psi \rangle^2}{2^N}, \qquad P \in \mathcal{P}_N. \tag{14}$$

Sampling Pauli strings according to the distribution $\pi(P)$ enables a Monte Carlo estimation of the SRE. Given a sample $\{P_k\}_{k=1}^{n_S}$ drawn (approximately) from π, a simple estimator of S_q in Eq. (11) is

$$\hat{S}_q(|\psi\rangle) \; = \; \frac{1}{n_S} \sum_{k=1}^{n_S} \langle \psi | P_k | \psi \rangle^{2(q-1)}, \tag{15}$$

see, e.g., [100], so that a sampling-based estimate of the SRE reads $\hat{M}_q(|\psi\rangle) = -\log_2[\hat{S}_q(|\psi\rangle)]$. The distribution $\pi(P)$ can be sampled, for example, directly with a Metropolis–Hastings algorithm [45], via Bell sampling [101], or using sampling approaches adapted to tensor network states [34, 100]. For large n_S, the central limit theorem gives $\delta S_q \sim \sigma_b / \sqrt{n_S}$, where $\sigma_b^2 = \text{Var}_\pi \big[\langle \psi | P | \psi \rangle^2 \big]$. Fixing $q > 1$, we see that the uncertainty on M_q is controlled by

$$\delta M_q \; \approx \; \frac{1}{\ln 2} \frac{\delta S_q}{S_q} \; \sim \; \frac{\sigma_b}{S_q \sqrt{n_S}}. \tag{16}$$

For non-stabilizer states, $S_q(|\psi\rangle)$ can decay exponentially with the number of qubits, so that the statistical error δM_q for fixed n_S can grow, in the worst-case scenario, exponentially with system size N. In such cases, achieving a fixed additive precision in M_q for any $q > 1$ requires a number of samples n_S that is exponential in N, rendering the naive sampling approach inefficient [102].

2. *Fast-Hadamard-enhanced sampling.*

The sampling of Eq. (14) treats all Pauli strings on equal footing. Here we exploit the fast Hadamard transform to accelerate the sampling procedure. The key observation is that, for a Pauli string parametrized as $P = X_{\mathbf{a}}Z_{\mathbf{b}}$, a single fast Hadamard transform over the Z-strings yields the expectation values $\langle\psi|X_{\mathbf{a}}Z_{\mathbf{b}}|\psi\rangle$ for all Pauli strings $X_{\mathbf{a}}Z_{\mathbf{b}}$ at once, for any fixed X-string $X_{\mathbf{a}}$, an idea already used in Algorithm 2.

Following similar approaches for mana [69] and entanglement [103–106], for each fixed $X_{\mathbf{a}}$ we define a regularized "energy"

$$f_\varepsilon(X_{\mathbf{a}}) \;=\; -\log\!\left(\sum_{\mathbf{b}\in\mathbb{Z}_2^N} \langle\psi|X_{\mathbf{a}}Z_{\mathbf{b}}|\psi\rangle^4 \;+\; \varepsilon\right), \qquad \varepsilon > 0, \tag{17}$$

so that the inner sum is strictly positive for all $\mathbf{a} \in \mathbb{Z}_2^N$. The parameter $\varepsilon \ll 1$ controls a small additive bias, ensuring that $f_\varepsilon(X_{\mathbf{a}})$ remains finite even if $\sum_{\mathbf{b}\in} \langle\psi|X_{\mathbf{a}}Z_{\mathbf{b}}|\psi\rangle^4$ vanishes for certain $\mathbf{a}$. To sample $X_{\mathbf{a}}$ we introduce the Boltzmann distribution

$$\Pi_\beta(X_{\mathbf{a}}) \;=\; \frac{1}{Z_\beta^{(\varepsilon)}}\, e^{-\beta f_\varepsilon(X_{\mathbf{a}})}, \qquad Z_\beta^{(\varepsilon)} = \sum_{\mathbf{a}} e^{-\beta f_\varepsilon(X_{\mathbf{a}})}, \tag{18}$$

with an auxiliary inverse temperature $\beta \in [0,1]$. The expectation value of f_ε with respect to Π_β is

$$\big\langle f_\varepsilon(X_{\mathbf{a}})\big\rangle_\beta = \sum_{\mathbf{a}\in\mathbb{Z}_2^N} \Pi_\beta(X_{\mathbf{a}})\, f_\varepsilon(X_{\mathbf{a}}) = -\frac{d}{d\beta}\log Z_\beta^{(\varepsilon)}, \tag{19}$$

and integrating over β yields the standard thermodynamic identity

$$\int_0^1 d\beta\, \big\langle f_\varepsilon(X_{\mathbf{a}})\big\rangle_\beta = \log Z_1^{(\varepsilon)} - \log Z_0^{(\varepsilon)}. \tag{20}$$

At $\beta = 1$ we obtain

$$Z_1^{(\varepsilon)} = \sum_{\mathbf{a}} e^{-f_\varepsilon(X_{\mathbf{a}})} = \sum_{\mathbf{a},\mathbf{b}\in\mathbb{Z}_2^N} \langle\psi|X_{\mathbf{a}}Z_{\mathbf{b}}|\psi\rangle^4 + 2^N\varepsilon = S_2(|\psi\rangle) + 2^N\varepsilon, \tag{21}$$

where we used that each Pauli string can be written uniquely as $X_{\mathbf{a}}Z_{\mathbf{b}}$. At $\beta = 0$, all Boltzmann weights are equal to unity and $Z_0^{(\varepsilon)} = 2^N$. Combining these relations with Eq. (20) and denoting

$$I_\varepsilon(\psi) \;=\; \int_0^1 d\beta\, \big\langle f_\varepsilon(X_{\mathbf{a}})\big\rangle_\beta \tag{22}$$

yields the SRE as

$$M_2(|\psi\rangle) = -\log_2\!\big(e^{I_\varepsilon(\psi)} - \varepsilon\big). \tag{23}$$

3. Monte Carlo estimator and sampling error.

In our numerical algorithm, to calculate the SRE, we discretize the integral (22) over β on a grid $\{\beta_\ell\}_{\ell=1}^L$ with quadrature weights $\{w_\ell\}$,

$$I_\varepsilon(\psi) \approx \hat{I}_\varepsilon(\psi) = \sum_{\ell=1}^L w_\ell \, \widehat{\langle f_\varepsilon \rangle}_{\beta_\ell}, \tag{24}$$

and estimate $\langle f_\varepsilon \rangle_{\beta_\ell}$ by running a Markov chain at fixed β_ℓ with stationary distribution $\Pi_{\beta_\ell}(X_{\mathbf{a}})$. If the chain at β_ℓ produces a series of samples $\{X_{\mathbf{a}}^{(n,\ell)}\}_{n=1}^{n_\ell}$, we use the sample mean

$$\widehat{\langle f_\varepsilon \rangle}_{\beta_\ell} = \frac{1}{n_\ell} \sum_{t=1}^{n_\ell} f_\varepsilon\big(X_{\mathbf{a}}^{(n,\ell)}\big). \tag{25}$$

For each β_ℓ, the central limit theorem for Markov chains implies that the variance of the sample mean $\widehat{\langle f_\varepsilon \rangle}_{\beta_\ell}$ scales as $2\,\tau_{\text{int}}(\beta_\ell)\sigma_f^2(\beta_\ell)/n_\ell$, where $\sigma_f^2(\beta_\ell)$ is the variance of $f_\varepsilon(X_{\mathbf{a}})$ under Π_{β_ℓ} and $\tau_{\text{int}}(\beta_\ell)$ is the corresponding integrated autocorrelation time. Since the runs for different β_ℓ are statistically independent, the variance of $\hat{I}_\varepsilon$ is

$$\text{Var}\big[\hat{I}_\varepsilon(\psi)\big] \approx \sum_{\ell=1}^L w_\ell^2 \, \frac{2\,\tau_{\text{int}}(\beta_\ell)}{n_\ell} \, \sigma_f^2(\beta_\ell). \tag{26}$$

The role of the regularization parameter ε is to ensure that the logarithm in $f_\varepsilon(X_{\mathbf{a}})$ is well-defined even when $S(\mathbf{a}) = \sum_{\mathbf{b}} \langle \psi | X_{\mathbf{a}} Z_{\mathbf{b}} | \psi \rangle^4$ vanishes for some $\mathbf{a}$. If we know a priori that $S(\mathbf{a}) > 0$ for all $\mathbf{a}$—as is empirically the case for the quantum-circuit states analyzed in Sec. IV—we may safely take the limit $\varepsilon \to 0$ in Eqs. (17) and (23) and work with $f(X_{\mathbf{a}}) = -\log S(\mathbf{a})$. In this regime $f(X_{\mathbf{a}})$ is finite for all $\mathbf{a}$. Denoting $s_{\min} = \min_{\mathbf{a}} S(\mathbf{a})$, the variance $\sigma_f^2(\beta_\ell)$ of $f(X_{\mathbf{a}})$ under Π_{β_ℓ} is then bounded from above by a constant of order $\log^2(1/s_{\min})$.

For the quantum-circuit states considered later in this work (Sec. IV), we empirically find $s_{\min}(N) \sim e^{-cN}$ with some constant $c > 0$. In this case $\log(1/s_{\min}) \sim cN$, so that the worst-case variance of $f(X_{\mathbf{a}})$ scales at most quadratically with system size, $\sigma_f^2(\beta_\ell) = O(N^2)$, and the corresponding contribution to $\text{Var}[\hat{I}(\psi)]$ in Eq. (26) grows at most polynomially in N.

For $\varepsilon = 0$, the stabilizer Rényi entropy in Eq. (23) can be estimated as $\hat{M}_2(|\psi\rangle) = -\hat{I}(\psi)/\ln 2$. Using the variance estimate

$$\text{Var}\big[\hat{I}(\psi)\big] \approx \sum_{\ell=1}^L w_\ell^2 \, \frac{2\,\tau_{\text{int}}(\beta_\ell)}{n_\ell} \, \sigma_f^2(\beta_\ell), \tag{27}$$

where n_ℓ is the number of Monte Carlo samples at β_ℓ, $\tau_{\text{int}}(\beta_\ell)$ the integrated autocorrelation time, and $\sigma_f^2(\beta_\ell)$ the variance of $f(X_{\mathbf{a}})$ under Π_{β_ℓ}, error propagation yields the statistical uncertainty on the SRE,

$$\delta M_2 \equiv \sqrt{\text{Var}\big[\hat{M}_2(|\psi\rangle)\big]} \approx \frac{1}{\ln 2} \sqrt{\sum_{\ell=1}^L w_\ell^2 \, \frac{2\,\tau_{\text{int}}(\beta_\ell)}{n_\ell} \, \sigma_f^2(\beta_\ell)}. \tag{28}$$

Equivalently, introducing an effective number of independent samples $n_{\text{eff}} \sim \sum_\ell n_\ell / \tau_{\text{int}}(\beta_\ell)$, we have the scaling $\delta M_2 \propto 1/\sqrt{n_{\text{eff}}}$, up to polynomial factors in N coming from $\sigma_f^2(\beta_\ell)$.

A polynomial growth of δM_2 with N is in stark contrast to the exponential increase of the statistical error in the naive sampling scheme, cf. Eq. (16). Consequently, for generic quantum states with $S(\mathbf{a}) > 0$, the present thermodynamic-integration approach requires only a polynomial increase of the total sampling effort (i.e. the number of Markov-chain updates $\{n_\ell\}$ at each β_ℓ) in order to maintain a fixed statistical precision δM_2 on the stabilizer Rényi entropy as the system size grows.

In the general case, $S(\mathbf{a})$ may vanish for some $\mathbf{a}$. In such cases the limit $\varepsilon \to 0$ is singular, and the error propagation yields

$$\delta M_2 \equiv \sqrt{\text{Var}\big[\hat{M}_2(|\psi\rangle)\big]} \approx \frac{1}{\ln 2}\left(1 + \frac{\varepsilon}{S_2(|\psi\rangle)/2^N}\right)\sqrt{\text{Var}\big[\hat{I}_\varepsilon(\psi)\big]}. \tag{29}$$

If $S_2(|\psi\rangle)/2^N \lesssim \varepsilon$, the prefactor becomes large, $1 + \varepsilon/(S_2/2^N) = O\big(\varepsilon/(S_2/2^N)\big)$, and the uncertainty δM_2 is strongly amplified by the finite value of ε. In this regime, resolving M_2 with fixed relative precision would require reducing ε in tandem with $S_2(|\psi\rangle)/2^N$, which in turn increases the variance of f_ε as $\sim \log^2(1/\varepsilon)$, and in the worst-case scenario may lead to an exponential increase of δM_2. However, the vanishing of $S(\mathbf{a})$ for a fixed set of $\{\mathbf{a}\}$ is a fine-tuned property of the quantum state $|\psi\rangle$, and does not occur in most of the practical applications of the sampling scheme.

4. Sampling scheme for SRE.

Algorithm 3. Sampling-based estimation of the SRE $M_2(|\psi\rangle)$.

Require: Number of qubits N; state $|\psi\rangle$; even integer L (number of grid points); number of samples n_S.

Set up a uniform grid $\beta_\ell = \ell\,\Delta\beta$, $\ell = 0, \ldots, L-1$, with $\Delta\beta = 1/(L-1)$, and Simpson weights w_ℓ.

for $\ell = 0$ to $L-1$ **do** $\triangleright$ Independent Markov chains at each β_ℓ

    Initialize a random X-pattern $X_{\mathbf{a}}^{(0,\ell)}$, and set $A_\ell \leftarrow 0$.

    for $n = 1$ to n_S **do**

        Propose a new pattern $X_{\mathbf{a}'}$ by a local move on $\mathbf{a}$.

        Compute $f(X_{\mathbf{a}'}) \leftarrow \text{EVALENERGY}(X_{\mathbf{a}'})$, and let $f(X_{\mathbf{a}}^{(n-1,\ell)})$ be the current value.

        Set $p_{\text{acc}} \leftarrow \min\big\{1, \exp\big[-\beta_\ell\big(f(X_{\mathbf{a}'}) - f(X_{\mathbf{a}}^{(n-1,\ell)})\big)\big]\big\}$.

        With probability p_{acc} set $X_{\mathbf{a}}^{(n,\ell)} \leftarrow X_{\mathbf{a}'}$; otherwise set $X_{\mathbf{a}}^{(n,\ell)} \leftarrow X_{\mathbf{a}}^{(n-1,\ell)}$.

        Accumulate $A_\ell \leftarrow A_\ell + f\big(X_{\mathbf{a}}^{(n,\ell)}\big)$.

    end for

    Estimate $\langle f\rangle_{\beta_\ell}$ as $\widehat{\langle f\rangle}_{\beta_\ell} \leftarrow A_\ell/n_S$.

end for

return $\hat{M}_2(|\psi\rangle) = -\frac{1}{\ln 2}\sum_{\ell=0}^{L-1} w_\ell \, \widehat{\langle f\rangle}_{\beta_\ell}$.

Here, for simplicity, we focus on the generic situation with $S(\mathbf{a}) > 0$ for all $\mathbf{a}$ and fix $\varepsilon = 0$. To sample the Boltzmann probability distribution $\Pi_\beta(X_{\mathbf{a}})$ in Eq. (18), we use the Metropolis–

Hastings algorithm [107, 108], which builds a Markov chain $\{X_{\mathbf{a}}^{(n)}\}_{n \geq 0}$ with stationary distribution Π_β. Starting from an initial pattern $X_{\mathbf{a}}^{(0)}$, each step consists of proposing a new pattern $X_{\mathbf{a}'}$ from a symmetric proposal distribution (e.g. by flipping a randomly chosen subset of bits in $\mathbf{a}$), evaluating the corresponding energy difference $\Delta f = f(X_{\mathbf{a}'}) - f(X_{\mathbf{a}}^{(n)})$, and accepting the move with probability

$$p_{\mathrm{acc}} = \min\{1, \exp[-\beta \Delta f]\},$$

otherwise keeping the current state. After an initial thermalization period, successive samples along the chain provide an ergodic sampling of $X_{\mathbf{a}}$ distributed according to Π_β, allowing us to estimate $\langle f \rangle_\beta$ by simple averaging over the sampled configurations.

The central computational step of our sampling scheme is the evaluation of the "energy" $f(X_{\mathbf{a}}) = -\log S(\mathbf{a})$, which we make efficient by exploiting the fast Hadamard transform. For a given X-pattern $X_{\mathbf{a}}$, the routine EVALENERGY first computes the amplitudes of the states $|\psi\rangle$ and $X_{\mathbf{a}} |\psi\rangle$ in the computational basis and forms a vector $v_{\mathbf{x}}$ whose components are suitable products of these amplitudes (as in Algorithm 2). A single fast Walsh–Hadamard transform applied to v then yields all overlaps $\chi_{\mathbf{b}} = \langle \psi | X_{\mathbf{a}} Z_{\mathbf{b}} | \psi \rangle$ for $\mathbf{b} \in \mathbb{Z}_2^N$ in one shot, in time $O(N2^N)$. The quantity $S(\mathbf{a}) = \sum_{\mathbf{b}} \chi_{\mathbf{b}}^4$ is obtained by a single pass over these overlaps, and EVALENERGY finally returns $f(X_{\mathbf{a}}) = -\log S(\mathbf{a})$. In this way, an explicit sum over all 4^N Pauli strings is replaced by a single fast transform plus linear-time post-processing. This procedure is repeated for each β on a uniform grid in $[0,1]$, generating n_S Monte Carlo samples at each β.

The overall complexity of the resulting Algorithm 3 is dominated by calls to EVALENERGY inside the Markov chains. Each evaluation costs $O(N2^N)$ operations, and it is invoked once per Monte Carlo update at each β_ℓ. Neglecting the thermalization phase, the dominating computational cost of the algorithm thus scales as $O\left(L n_S N 2^N\right)$. For fixed grid size L and polynomially growing sample numbers n_ℓ, the algorithm is therefore quasi-linear in the Hilbert-space dimension 2^N, in sharp contrast to any scheme that would explicitly iterate over all 4^N Pauli strings.

C. Calculation of mana for pure qutrit states

1. Naive mana calculation

The close formal resemblance between Eqs. (2) and (10) implies that the naive Gray-code sweep used for the SRE in Algorithm 1 can be adapted to compute mana with only minor modifications. For a pure state $|\psi\rangle$ of N qutrits, the mana $\mathcal{M}(|\psi\rangle)$ is obtained from the discrete Wigner function and requires evaluating the 9^N expectation values $\langle \psi | A_{\mathbf{u}} | \psi \rangle$, where $\{A_{\mathbf{u}}\}$ denotes the set of phase-space point operators labeled by $\mathbf{u} \in \mathbb{Z}_3^{2N}$.

In the following, we set $\omega \equiv \omega_3 = e^{i2\pi/3}$ and reparameterize $\mathbf{u}$ as a pair of N-component ternary strings $(\mathbf{a}, \mathbf{b}) \in \mathbb{Z}_3^N \times \mathbb{Z}_3^N$, defining

$$A_{\mathbf{ab}} = \Big(\prod_{j=1}^N X_j^{a_j} Z_j^{b_j}\Big) A_{\mathbf{0}} \Big(\prod_{j=1}^N X_j^{a_j} Z_j^{b_j}\Big)^\dagger. \tag{30}$$

Because $X_j^a Z_j^b = \omega^{r(a,b)} Z_j^b X_j^a$ (with $r(a,b) \in \{0,1,2\}$), this definition is equivalent to Eq. (7) up to an overall phase. Moreover, a direct calculation shows that the origin operator acts as a reflection

in the computational basis,

$$A_{\mathbf{0}} \ket{\mathbf{x}} = \ket{-\mathbf{x}}, \tag{31}$$

with $\mathbf{x} \in \mathbb{Z}_3^N$ and subtraction taken modulo 3 [69].

A straightforward (but costly) approach to compute mana is therefore to iterate over all $(\mathbf{a}, \mathbf{b}) \in \mathbb{Z}_3^N \times \mathbb{Z}_3^N$, construct $A_{\mathbf{ab}} \ket{\psi}$, and evaluate the overlaps $\bra{\psi} A_{\mathbf{ab}} \ket{\psi}$. As in the qubit case, one can order the updates using a ternary Gray code for $\mathbf{a}$ and $\mathbf{b}$ (so that successive strings differ at a single position), resulting in Algorithm 4. The memory cost of this scheme is $O(3^N)$, set by storing state vectors of N qutrits, while the runtime is dominated by the evaluation of the 9^N overlaps, each involving $O(3^N)$ operations. This leads to an overall computational complexity $O(27^N)$, which grows exponentially with system size and restricts this naive approach to relatively small N.

Algorithm 4. Mana calculation via a Gray-code sweep of phase-space point operators.

Require: Number of qutrits N; state $\ket{\psi}$.
  $m \leftarrow 0$; $\mathbf{a} \leftarrow (0, \ldots, 0)$, $\mathbf{b} \leftarrow (0, \ldots, 0)$; $\ket{\phi} \leftarrow A_{\mathbf{0}} \ket{\psi}$. $\triangleright$ Initialize at $(\mathbf{a}, \mathbf{b}) = (\mathbf{0}, \mathbf{0})$.
  for $k = 0$ to $3^N - 1$ **do** $\triangleright$ Loop over $\mathbf{a} \in \mathbb{Z}_3^N$.
    for $l = 0$ to $3^N - 1$ **do** $\triangleright$ Loop over $\mathbf{b} \in \mathbb{Z}_3^N$.
      $m \leftarrow m + \left| \braket{\psi | \phi} \right|$.
      $\mathbf{b}, t \leftarrow \text{NextGray}_3(\mathbf{b})$. $\triangleright$ Ternary Gray step; t is the position incremented.
      $\ket{\phi} \leftarrow X_t \ket{\phi}$. $\triangleright$ Update corresponding to $\mathbf{b} \to \mathbf{b}'$.
    end for
    $\mathbf{a}, w \leftarrow \text{NextGray}_3(\mathbf{a})$. $\triangleright$ Advance ternary Gray code for $\mathbf{a}$.
    $\ket{\phi} \leftarrow X_w \ket{\phi}$. $\triangleright$ Update corresponding to $\mathbf{a} \to \mathbf{a}'$.
  end for
  return $\mathcal{M}(\ket{\psi}) = \log_2(m)$

---

2. Fast Hadamard transform for mana

As in the qubit case, an exponential speedup over the naive Gray-code sweep for mana hinges on accelerating the inner loop over the "Z-type" phase-space index. Recall that mana for N qutrits requires evaluating the overlaps $\bra{\psi} A_{\mathbf{ab}} \ket{\psi}$ for all $(\mathbf{a}, \mathbf{b}) \in \mathbb{Z}_3^N \times \mathbb{Z}_3^N$. For a fixed $\mathbf{a}$, we define the $\mathbf{a}$-dependent state

$$\ket{\psi_{\mathbf{a}}} \equiv \left(\prod_{j=1}^N X_j^{a_j} \right) \ket{\psi}. \tag{32}$$

Then, up to an overall phase that is irrelevant for the absolute value entering mana, the inner sweep over $\mathbf{b}$ amounts to evaluating the family of overlaps

$$\bra{\psi} A_{\mathbf{ab}} \ket{\psi} = \bra{\psi_{\mathbf{a}}} Z_{\mathbf{b}} A_{\mathbf{0}} Z_{\mathbf{b}}^{\dagger} \ket{\psi_{\mathbf{a}}}, \qquad Z_{\mathbf{b}} \equiv \prod_{j=1}^N Z_j^{b_j}, \tag{33}$$

for all $\mathbf{b} \in \mathbb{Z}_3^N$. Using $A_{\mathbf{0}} |\mathbf{x}\rangle = |-\mathbf{x}\rangle$ (with arithmetic modulo 3) [69], these overlaps can be reduced to a ternary Hadamard-type transform, i.e., a Fourier transform over the additive group $\mathbb{Z}_3^N$.

To see this, we expand the state in the computational basis as $|\psi_{\mathbf{a}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_3^N} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$. Using $A_{\mathbf{0}} |\mathbf{x}\rangle = |-\mathbf{x}\rangle$ and $Z_{\mathbf{b}} |\mathbf{x}\rangle = \omega^{\mathbf{b} \cdot \mathbf{x}} |\mathbf{x}\rangle$, a direct computation yields

$$\chi_{\mathbf{b}}(\mathbf{a}) \equiv \langle\psi_{\mathbf{a}}| Z_{\mathbf{b}} A_{\mathbf{0}} Z_{\mathbf{b}}^{\dagger} |\psi_{\mathbf{a}}\rangle = \sum_{\mathbf{x} \in \mathbb{Z}_3^N} \alpha_{\mathbf{x}}^* \alpha_{-\mathbf{x}} \omega^{2\,\mathbf{b} \cdot \mathbf{x}}, \qquad \mathbf{b} \in \mathbb{Z}_3^N, \tag{34}$$

with all arithmetic performed modulo 3. Thus, for each fixed $\mathbf{a}$, the full set of overlaps $\{\chi_{\mathbf{b}}(\mathbf{a})\}_{\mathbf{b} \in \mathbb{Z}_3^N}$ is obtained by applying a tensor-product Fourier transform to the vector $v_{\mathbf{x}} = \alpha_{\mathbf{x}}^* \alpha_{-\mathbf{x}}$.

To make the analogy with the qubit relation $(\chi_{\mathbf{b}})_{\mathbf{b}} = H^{\otimes N}(v_{\mathbf{x}})_{\mathbf{x}}$ explicit, introduce the unnormalized 3×3 "Hadamard" matrix

$$(F_3)_{j,k} = \omega^{2jk}, \qquad j, k \in \{0, 1, 2\}, \tag{35}$$

and consider its tensor power $F_3^{\otimes N}$ acting on vectors indexed by $\mathbb{Z}_3^N$. Then Eq. (34) can be written simply as

$$\left(\chi_{\mathbf{b}}(\mathbf{a})\right)_{\mathbf{b} \in \mathbb{Z}_3^N} = F_3^{\otimes N}\left(v_{\mathbf{x}}\right)_{\mathbf{x} \in \mathbb{Z}_3^N}, \tag{36}$$

which is the direct qutrit analogue of the qubit formula for SRE (13) with $H^{\otimes N}$ replaced by $F_3^{\otimes N}$.

Algorithm 5. Efficient calculation of mana using a fast $\mathbb{Z}_3^N$ Fourier transform.

Require: Number of qutrits N; state $|\psi\rangle$.
  $m \leftarrow 0$, $\mathbf{a} \leftarrow (0, \ldots, 0)$, $|\psi_{\mathbf{a}}\rangle \leftarrow |\psi\rangle$.
  for $k = 0$ to $3^N - 1$ **do**   ▷ Loop over $\mathbf{a} \in \mathbb{Z}_3^N$.
    Extract amplitudes $\alpha_{\mathbf{x}}$ of $|\psi_{\mathbf{a}}\rangle = \sum_{\mathbf{x}} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$.
    Form $v_{\mathbf{x}} \leftarrow \alpha_{\mathbf{x}}^* \alpha_{-\mathbf{x}}$ for all $\mathbf{x} \in \mathbb{Z}_3^N$.
    $(\chi_{\mathbf{b}})_{\mathbf{b}} \leftarrow \text{FASTFT}_{\mathbb{Z}_3^N}\left((v_{\mathbf{x}})_{\mathbf{x}}\right)$ ▷ Compute Eq. (34).
    $m \leftarrow m + \sum_{\mathbf{b} \in \mathbb{Z}_3^N} |\chi_{\mathbf{b}}|$
    Update $\mathbf{a} \leftarrow \text{NEXTGRAY}_3(\mathbf{a})$; update $|\psi_{\mathbf{a}}\rangle$ accordingly.
  end for
  return $\mathcal{M}(|\psi\rangle) = \log_2(m)$

Algorithm 5 implements this strategy by replacing the explicit sweep over all phase-space point operators with a fast $\mathbb{Z}_3^N$ Fourier transform for each fixed $\mathbf{a}$. In this way, the cost of the inner loop is reduced from $O(9^N)$ to $O(N3^N)$, while the outer loop over $\mathbf{a}$ still involves $O(3^N)$ iterations. The resulting time complexity is $O(N9^N)$, providing an exponential speedup over the naive $O(27^N)$ approach of Algorithm 4. The memory footprint remains $O(3^N)$, set by storing the state vector $|\psi_{\mathbf{a}}\rangle$ and work arrays of length 3^N.

D. Mana of mixed states via a fast tensor-product transform

When the target state is specified by a density matrix ρ (possibly mixed), the computational strategy must be modified, since expectation values can no longer be reduced to state-vector overlaps.

For qubits, the stabilizer Rényi entropy is a magic monotone *only* for pure-state stabilizer protocols [32]. Nevertheless, the SRE can be used to construct a magic *witness* for mixed states [109]. To compute the SRE for a mixed state ρ [34] of N qubits, one can proceed by calculating the full set of Pauli expectation values $\mathrm{Tr}(P_j \rho)$, as summarized in Appendix B.1 of Ref. [30]. This yields an in-place Pauli-decomposition algorithm with runtime $O(N4^N)$ and memory cost $O(4^N)$, improving over the naive $O(8^N)$ sweep. Closely related routines for Pauli decomposition of dense operators have also been developed in Refs. [110–112].

In contrast to SRE, mana in qutrit systems remains a faithful non-stabilizerness monotone for mixed states [18], making its calculation for mixed states particularly relevant. Accordingly, below we present an exact fast-transform algorithm that efficiently evaluates the full set of phase-space expectation values $\mathrm{Tr}(\rho A_{\mathbf{u}})$ and thereby enables the computation of mana for mixed states.

For an N-qutrit density matrix ρ, mana $\mathcal{M}(\rho)$ is obtained from the discrete Wigner function $W_{\mathbf{u}} = \mathrm{Tr}(\rho A_{\mathbf{u}})$ by summing $|W_{\mathbf{u}}|$ over all 9^N phase-space points $\mathbf{u} \in \mathbb{Z}_3^{2N}$, cf. Eq. (9). In contrast to the pure-state setting, where overlaps $\langle \psi | A_{\mathbf{u}} | \psi \rangle$ can be generated from a state vector, in the mixed-state setting one must evaluate $\mathrm{Tr}(\rho A_{\mathbf{u}})$ for all $\mathbf{u} \in \mathbb{Z}_3^{2N}$. A naive calculation of the 9^N traces yields a computational cost $O(3^{3N})$. The key observation enabling an exponential reduction of the computational cost is that the entire collection $\{\mathrm{Tr}(\rho A_{\mathbf{u}})\}_{\mathbf{u} \in \mathbb{Z}_2^{2N}}$ can be written as a single linear map acting on a vectorized density matrix, and, crucially, this map factorizes as a tensor power of a fixed 9×9 single-qutrit matrix. This structure enables an exact fast tensor-product transform implementation of $\mathcal{M}(\rho)$ calculation that utilizes $M^{\otimes N}$ without ever forming a $9^N \times 9^N$ matrix.

1. *Single-qutrit construction.*

We collect the expectation values of the phase-space point operators in a vector $w \in \mathbb{C}^9$ with components

$$w_u = \mathrm{Tr}(\rho A_u), \qquad u \in \mathbb{Z}_3^2, \tag{37}$$

and represent the density matrix ρ in the column-stacking convention

$$\mathrm{vec}(\rho) = \left(\rho_{00}, \rho_{10}, \rho_{20}, \rho_{01}, \rho_{11}, \rho_{21}, \rho_{02}, \rho_{12}, \rho_{22}\right)^{\mathsf{T}} \in \mathbb{C}^9. \tag{38}$$

Using $\mathrm{Tr}(\rho A_u) = \mathrm{vec}(A_u^{\mathsf{T}})^{\mathsf{T}} \mathrm{vec}(\rho)$, we can write

$$w = M \, \mathrm{vec}(\rho), \tag{39}$$

where the 9×9 matrix M is defined as

$$M_{u,\alpha} = \left[\mathrm{vec}(A_u^{\mathsf{T}})\right]_\alpha, \qquad u \in \mathbb{Z}_3^2, \ \alpha \in \{0, \dots, 8\}. \tag{40}$$

The matrix M has a particularly simple structure. In the $(u = (a, a'))$ row and (q, p) column labelling induced by the vectorization above, each row of M has exactly three non-zero entries, with values in $\{1, \omega, \omega^2\}$. More precisely, for fixed a' the three rows with $a = 0, 1, 2$ have support on a common "line" of three operator-basis elements (q, p) satisfying $q + p \equiv -a' \pmod 3$, and on that line the 3×3 submatrix is (up to permutations) the qutrit matrix F_3 given by Eq. (35). Accordingly, there exist permutation matrices P_{row} and P_{col} such that

$$M = P_{\mathrm{row}}^{\mathsf{T}} \left(F_3 \oplus F_3 \oplus F_3\right) P_{\mathrm{col}}, \tag{41}$$

i.e. in a suitably ordered basis the phase-space transform on a single qutrit decomposes as a direct sum of three identical 3×3 Fourier blocks.

2. Construction for many qutrits.

Let ρ be a density operator for system of N qutrits, with computational basis $\{|\mathbf{i}\rangle\}_{\mathbf{i}\in\mathbb{Z}_3^N}$, $\mathbf{i} = (i_0, \ldots, i_{N-1})$. We write matrix elements as $\rho_{\mathbf{i},\mathbf{j}} = \langle\mathbf{i}|\rho|\mathbf{j}\rangle$, and regard ρ as a tensor with indices $(i_0, \ldots, i_{N-1}; j_0, \ldots, j_{N-1})$. At each site k, we combine the bra/ket indices into a single local label

$$\mu_k \equiv 3i_k + j_k \in \{0, \ldots, 8\}. \tag{42}$$

This identifies the local operator space with $\mathbb{C}^9$ spanned by operators $\{|\mu_k\rangle\}_{\mu_k=0}^8$, and we define Vec_N as mapping from the operator space $\mathcal{L}\big((\mathbb{C}^3)^{\otimes N}\big)$ to vector space $(\mathbb{C}^9)^{\otimes N}$ by

$$|\mathbf{i}\rangle\langle\mathbf{j}| \longmapsto |\mu_0\rangle \otimes \cdots \otimes |\mu_{N-1}\rangle. \tag{43}$$

Equivalently, the resulting vector $v = \mathrm{Vec}_N(\rho) \in \mathbb{C}^{9^N}$ is indexed by the base-9 integer

$$p(\mathbf{i}, \mathbf{j}) = \sum_{k=0}^{N-1} \mu_k \, 9^{N-1-k}, \tag{44}$$

so that $v_{p(\mathbf{i},\mathbf{j})} = \rho_{\mathbf{i},\mathbf{j}}$. This convention ensures that product operators factorize as $\mathrm{Vec}_N(B^{(1)} \otimes \cdots \otimes B^{(N)}) = \mathrm{vec}(B^{(1)}) \otimes \cdots \otimes \mathrm{vec}(B^{(N)})$.

Algorithm 6. Exact computation of mixed-state mana via application of $M^{\otimes N}$.

Require: Number of qutrits N; density matrix $\rho \in \mathbb{C}^{3^N \times 3^N}$ stored column-major.

Construct $M \in \mathbb{C}^{9 \times 9}$ row-wise via $M_{u,:} = \big(\vec{A}_u^{\mathsf{T}}\big)^{\mathsf{T}}$ (column-major vec).

Compute $v = \mathrm{Vec}_N(\rho) \in \mathbb{C}^{9^N}$ as follows:

for $c = 0$ to $3^N - 1$ **do** $\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ ▷ column index of ρ

$\quad$ Decode c into base-3 digits $\mathbf{j} = (j_0, \ldots, j_{N-1})$.

$\quad$ **for** $r = 0$ to $3^N - 1$ **do** $\qquad\qquad\qquad\qquad\qquad\qquad$ ▷ row index of ρ

$\quad\quad$ Decode r into base-3 digits $\mathbf{i} = (i_0, \ldots, i_{N-1})$.

$\quad\quad$ Set $\mu_k = 3i_k + j_k$ and initialize $p \leftarrow 0$; then update $p \leftarrow 9p + \mu_k$ for $k = 0, \ldots, N-1$.

$\quad\quad$ Assign $v[p] \leftarrow \rho_{r,c}$ (column-major: $\rho_{r,c} = \rho[r + c\,3^N]$).

$\quad$ **end for**

end for

Apply $M^{\otimes N}$ in place (leg sweep):

for $\ell = 0$ to $N - 1$ **do** $\qquad\qquad\qquad\qquad$ ▷ $\ell = 0$ is the most-significant base-9 digit

$\quad$ $n_s \leftarrow 9^{N-1-\ell}$, $b_s \leftarrow 9\,n_s$, $n_b \leftarrow 9^N/b_s$

$\quad$ **for** $b = 0$ to $n_b - 1$ **do**

$\quad\quad$ $\mathrm{base} \leftarrow b \cdot b_s$.

$\quad\quad$ **for** $m = 0$ to $n_s - 1$ **do** $\qquad\qquad\qquad\qquad$ ▷ m labels fibers within the block

$\quad\quad\quad$ Gather the fiber along leg ℓ: $\quad x_a \leftarrow v[\mathrm{base} + m + a\,n_s]$ for $a = 0, \ldots, 8$.

$\quad\quad\quad$ Compute $y \leftarrow Mx$ (dense 9×9 matrix-vector product).

$\quad\quad\quad$ Write back $v[\mathrm{base} + m + a\,n_s] \leftarrow y_a$ for $a = 0, \ldots, 8$.

$\quad\quad$ **end for**

$\quad$ **end for**

end for

Now $v = w$ with entries $w_{\mathbf{u}} = \mathrm{Tr}(\rho A_{\mathbf{u}})$ in the base-9 ordering of $\mathbf{u}$.

return $\mathcal{M}(\rho) = \log_2 \sum_{\mathbf{u}} |w_{\mathbf{u}}|$ consistently with Eq. (9).

For any phase-space point operator $A_{\mathbf{u}} = A_{u_0} \otimes \cdots \otimes A_{u_{N-1}}$ with $\mathbf{u} \in \mathbb{Z}_3^{2N}$, the vector of the expectation values $w_{\mathbf{u}} = \text{Tr}(\rho A_{\mathbf{u}})$, ordered consistently with the base-9 encoding of $\mathbf{u}$, can be calculated as

$$w = M^{\otimes N} \text{Vec}_N(\rho), \tag{45}$$

because both the operator basis and the vectorization respect tensor products. Thus, computing all 9^N expectation values reduces to applying $M^{\otimes N}$ to a length-9^N vector, in an analogy to the structure of SRE and mana computation for pure states unravelled by Eq. (13) and (36), respectively.

To efficiently implement Eq. (45), in Algorithm 6, we apply $M^{\otimes N}$ *in place* by sweeping over the tensor indices ("legs"), each corresponding to one qutrit. We regard the vector $v \in \mathbb{C}^{9^N}$ as an N-index array with local dimension 9, i.e. as a tensor $v_{\mu_0,\ldots,\mu_{N-1}}$. For a fixed leg $\ell = 0, \ldots, N-1$, we hold all other indices $\mu_{k \neq \ell}$ fixed and apply the single-qutrit matrix M along the remaining index μ_ℓ. The corresponding set of nine entries $\{v_{\mu_0,\ldots,\mu_\ell,\ldots,\mu_{N-1}}\}_{\mu_\ell=0}^{8}$ is what we refer to as a *fiber* along leg ℓ. In our memory layout, these fibers appear as nine equally spaced elements in the flat array. Writing indices in base 9, entries that differ only in the ℓ-th digit are separated by $n_s = 9^{N-1-\ell}$ in memory. Each contiguous block of length $9n_s$ contains all fibers obtained by varying μ_ℓ while holding the remaining indices fixed. For every block (with starting index base) and every offset $m \in \{0, \ldots, n_s - 1\}$, we gather the nine elements $\{v[\text{base} + m + a\,n_s]\}_{a=0}^{8}$ into a local vector x, compute $y = Mx$, and write the result back to the corresponding locations. Sweeping over all offsets m and all blocks in this way processes every fiber exactly once for each ℓ, and thus realizes the full action of $M^{\otimes N}$ without ever forming the $9^N \times 9^N$ matrix explicitly.

The cost per leg is the number of fibers (9^{N-1}) times the constant cost of a dense 9×9 matrix-vector multiplication, giving $O(9^N)$ operations per leg. This results in a total computational complexity $O(N\,9^N)$ of the Algorithm 6. The memory footprint is $O(9^N)$, dominated by storing the density matrix $v = \text{Vec}_N(\rho)$.

IV. EXAMPLES AND BENCHMARKS

In this section we benchmark Algorithms 2, 3, 5, and 6, as implemented in HadaMAG.jl , on states generated by random quantum circuits. Such circuits provide a canonical source of highly entangled many-body states, for which state-vector methods are the natural tools of choice. We consider a one-dimensional chain of N qubits ($d = 2$) or qutrits ($d = 3$), evolved by a brick-wall circuit. The evolution operator is $U_t = \prod_{r=1}^{t} U^{(r)}$, where t is the circuit depth (or time), with layers

$$U^{(2m)} = \prod_{i=1}^{N/2-1} u_{2i,2i+1};, \quad U^{(2m+1)} = \prod_{i=1}^{N/2} u_{2i-1,2i};, \tag{46}$$

where each two-qudit gate $u_{i,j}$ is drawn from the unitary group $\mathcal{U}(d^2)$ with respect to the Haar measure.

The SRE and mana routines in HadaMAG.jl support several backends—`:serial`, `:threads`, `:mpi_threads`, `:cuda`, and `:cuda_threads`—which exploit different levels of parallelism, as illustrated in the examples below and detailed in the documentation.

A. Exact calculation of stabilizer Rényi entropy

We begin by benchmarking the numerically exact computation of SRE for qubits. Starting from the product state $|0\rangle^{\otimes N}$, we generate a Haar-random brick-wall circuit state by applying Eq. (46), implemented in HadaMAG.jl as psi = rand_haar(N; depth = t). We then evaluate the stabilizer Rényi entropy $M_2(|\psi\rangle)$ with Algorithm 2 using the call SRE(psi, 2).

```julia
julia> using HadaMAG

julia> psi = rand_haar(16; depth=0)
StateVec{ComplexF64,2}(n=16, dim=65536, mem=1.0 MiB)

julia> m,l=SRE(psi, 2); println("SRE=\$m  lost_norm=\$l")
SRE=-0.0  lost_norm=0.0)

julia> psi=rand_haar(16; depth=1); m,l=SRE(psi, 2); println("SRE=\$m  lost_norm=\$l")
SRE=5.519381245826574 lost_norm=2.220446049250313e-15

julia> psi=rand_haar(16; depth=16); m,l=SRE(psi, 2);println("SRE=\$m  lost_norm=\$l")
SRE=13.985447072668483 lost_norm=3.3306690738754696e-15
```

The initial state $|0\rangle^{\otimes N}$ is a stabilizer state and, accordingly, we find $M_2(|0\rangle^{\otimes N}) = 0$. As the circuit depth t increases, the SRE grows monotonically and, for large t, approaches the value characteristic of Haar-random states, $M_2^{\text{Haar}} = \log_2(2^N + 3) - 2$ [113]. In the dynamics of random circuits, this saturation to the Haar value occurs at circuit depths that scale only logarithmically with the system size N, as analyzed quantitatively in Ref. [46]. The second number lost_norm returned by SRE routine, is equal to $1 - \sum_{P \in \mathcal{P}_N} \langle \psi | P | \psi \rangle^2 / 2^N$, which for a normalized state $|\psi\rangle$ should be numerically equal to 0, cf., Eq. (14). To exploit parallelism in HadaMAG.jl, it suffices to select an appropriate backend in the SRE routine. For instance, multithreading within a single compute node is enabled by

```julia
julia> using HadaMAG

julia> psi = rand_haar(16; depth=4)
StateVec{ComplexF64,2}(n=16, dim=65536, mem=1.0 MiB)

julia> m,l = SRE(psi, 2; backend=:threads); println("SRE=\$m  lost_norm=\$l")
SRE=11.120916275249016 lost_norm=2.9976021664879227e-15
```

where the number of threads is controlled by the environment variable JULIA_NUM_THREADS. In Algorithm 2, parallelization is achieved by partitioning the loop over the binary Gray code into chunks, each associated with a distinct set of X-strings. Since these chunks are independent, the algorithm exhibits a high degree of parallelism and yields nearly ideal speedup as the number n_{cores} of cores is increased.

The performance of HadaMAG.jl on a single node with $n_{\text{cores}} = 112$ is summarized in Table I, and allows one to compute the SRE for systems up to $N = 22$ qubits in under one hour. For comparison, an implementation of the naive Algorithm 1 is limited to $N = 16$ qubits within the same wall time on a single node, illustrating the exponential speedup provided by the fast Hadamard transform.

The package HadaMAG.jl also supports MPI-based parallelism via the `:mpi_threads` backend; see the documentation for details. In addition, we provide efficient CUDA kernels for Algorithm 2, accessible through the `:cuda` and `:mpi_cuda` backends. For example, computing the SRE with HadaMAG.jl on multiple GPUs with MPI communication can be done as follows:

```julia
using HadaMAG
using CUDA
using MPI
using Random

MPI.Initialized() || MPI.Init()
comm = MPI.COMM_WORLD; rank = MPI.Comm_rank(comm)

Random.seed!(123); psi = rand_haar(L; depth=5)

batch_size = 128; nthreads_per_device = 128
m2, lost_norm = SRE(psi, 2; backend=:mpi_cuda, progress=false,batch=batch_size,
    threads=nthreads_per_device)
```

Table I summarizes benchmarks for the MPI-based backends on $n_{\mathrm{nodes}} = 8$ compute nodes, each equipped with 112 CPU cores, and compares them with SRE computations performed on 8 nodes where each node uses 4 GPUs. Offloading the core kernels to GPUs yields a substantial performance gain for system sizes $N \gtrsim 18$ qubits, allowing us to compute the SRE for $N = 25$ qubits in roughly 2 hours of wall-clock time.

We note that the reported memory usage is significantly larger than the memory required to store the state vector and a single set of auxiliary arrays: each thread needs its own auxiliary workspace for the fast Hadamard transform, so the total memory consumption grows with the number of threads. Although alternative designs are possible, in HadaMAG.jl we have adopted this choice to minimize total runtime; for the system sizes of interest, the resulting memory footprint remains well within the capabilities of current computing architectures.

B.   Sampling algorithm for qubit SRE

The $O(N4^N)$ scaling of Algorithm 2, reflected in the total CPU/GPU times reported in Table I, implies that the computational cost of *exact* SRE evaluation reaches thousands of CPU hours already for $N = 24$. By contrast, applying a single two-qubit gate from the circuit Eq. (46) costs only $O(2^N)$, so computing time-evolved state vectors of size $N \approx 30$ qubits is well within reach on contemporary hardware. In this regime, Algorithm 3 becomes a natural method of choice for *approximate* SRE estimation: its cost scales as $O(n_S N 2^N)$, where n_S is the number of Monte Carlo samples. Below, we compare the numerically exact SRE for $N = 16$ qubits with the estimate obtained from Algorithm 3 using $n_S = 1000$.

	N	t_w	$t_{\mathrm{CPU}}^{\mathrm{tot}}$	n_{cores}	RAM [MB]
naive	12	0.73 s	1.37 min	112	0.13
	14	41.4 s	1.29 h	112	0.32
	16	44.2 min	82.5 h	112	1.07
HadaMAG.jl	12	18 ms	2.02 s	112	5
	14	32 ms	3.63 s	112	17
	16	0.32 s	35.9 s	112	68
	18	4.83 s	9.01 min	112	269
	20	2.66 min	4.96 h	112	1074
	22	55.0 min	103 h	112	4295

	N	n_{nodes}	t_w	t^{tot}	RAM [MB]
SRE (GPU)	18	8	0.78 s	24.96 s	30
	20	8	8.23 s	4.39 min	120
	22	8	1.87 min	59.84 min	481
	24	8	0.06 min	1.92 min	1929
	25	8	2.10 h	67.20 h	3866
SRE (CPU)	18	8	0.76 s	11.35 min	43
	20	8	21.46 s	5.34 h	174
	22	8	6.88 min	102.74 h	694
	24	8	2.24 h	2007 h	2778
	25	8	9.44 h	8458 h	5556

TABLE I. Performance of SRE computation in HadaMAG.jl. **Left:** CPU-only runs, comparing a custom implementation of Algorithm 1 ("naive") with Algorithm 2 implemented in HadaMAG.jl. For each system size N, we report the total wall-clock time t_w, the aggregate CPU time $t_{\mathrm{CPU}}^{\mathrm{tot}}$ summed over all $N_{\mathrm{cores}} = 112$ cores of a single node, and the peak memory usage. The CPU used: Intel Sapphire Rapid 8480+, 112 cores/node). **Right:** Comparison between the CPU and GPU backends of HadaMAG.jl for $N = 18$–25. Each job uses $n_{\mathrm{nodes}} = 8$ nodes, corresponding to 8×112 CPU cores and 8×4 GPUs, respectively; we report the wall-clock time t_w, the total device time t^{tot} (CPU-core hours for the CPU runs, GPU-device hours for the GPU runs), and peak RAM. The GPUs used: Nvidia Hopper with 64 HBM2.

```julia
julia> using HadaMAG

julia> L = 16; psi = rand_haar(L; depth=2)
StateVec{ComplexF64,2}(n=16, dim=65536, mem=1.0 MiB)

julia> m,l=SRE(psi, 2); println("SRE=\$m  lost_norm=\$l")
SRE=14.00006877485478  lost_norm=2.460254222569347e-13)

julia> m_approx=MC_SRE(psi, 2; Nsamples=1000); println("SRE_approx=\$m_approx")
SRE_approx=14.000013644355313
```

In the example above, the difference between the exact value of the SRE and its Monte Carlo estimate is approximately 5.5×10^{-5}. This small discrepancy reflects the fact that the variance σ_f^2 of the "energy" $f(X_{\mathbf{a}})$ appearing in Eq. (17)—to which the statistical error δM_2 is proportional, cf. Eq. (28)—is particularly small for states close to Haar-random, such as those generated at circuit depths $t \approx N$.

The variance σ_f^2, which sets the number of samples n_S required to reach a given precision in the SRE estimate, depends on the state under consideration. In Fig. 1 we analyze its behavior for states generated by random quantum circuits. We find that, beyond a threshold system size, σ_f is well approximated by a linear function of N. At fixed N, the variance decreases monotonically with increasing circuit depth t.[114] This linear growth of σ_f translates into a relatively mild, quadratic increase in the number of samples n_S needed to maintain a fixed statistical error on the SRE estimate as N grows. The right panel of Fig. 1 illustrates the convergence of the sampling algorithm towards the numerically exact value of M_2 as the sample number n_S is increased.
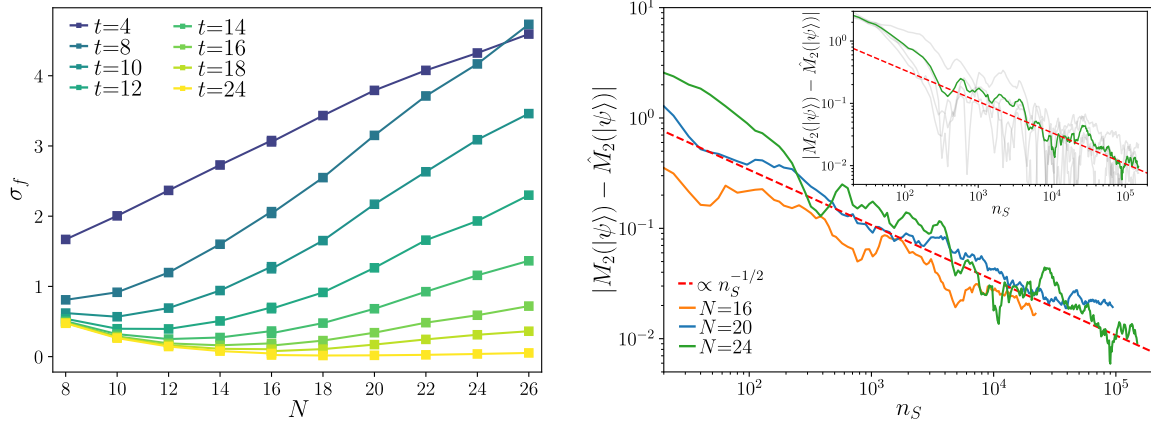
FIG. 1. **Left:** Variance σ_f of the "energy" $f(X_{\mathbf{a}})$ at $\beta = 1$, which controls the statistical uncertainty of the SRE in the sampling Algorithm 3, for states evolved under quantum circuits (46) of depth t, plotted as a function of system size N. **Right:** Absolute error of the SRE M_2 at circuit depth $t = 4$, comparing the numerically exact result from Algorithm 2 with its sampling estimate $\hat{M}_2$, as a function of the number of samples n_S. The data are averaged over four independent realizations of the sampling procedure. The inset shows the absolute SRE error (in grey) for each of the four realizations and their average for $N = 24$. Consistently with Eq. (28), the absolute error decreases as $n_S^{-1/2}$, as indicated by the red dashed lines.

C. Mana calculation

Computation of mana in HadaMAG.jl follows a pattern similar to that used for the SRE. We begin by benchmarking the numerically exact evaluation of mana for qutrits using Algorithm 5. Starting from the product state $|0\rangle^{\otimes N}$, we generate a Haar-random brick-wall circuit state by applying Eq. (46), implemented for qutrits as $\mathsf{psi} = \mathsf{rand\_haar(N;\ depth\ =\ t,\ q\ =\ 3)}$, and then compute its mana $\mathcal{M}(|\psi\rangle)$:

```julia
julia> using HadaMAG

julia> psi = rand_haar(10; depth=0, q=3)
StateVec{ComplexF64,3}(n=10, dim=59049, mem=923.0 KiB)

julia> Mana(psi)
[===============================================] 100.0%  (59049/59049)
-1.9706086672703374e-14

julia> psi = rand_haar(10; depth=4, q=3)
StateVec{ComplexF64,3}(n=10, dim=59049, mem=923.0 KiB)

julia> Mana(psi)
[===============================================] 100.0%  (59049/59049)
4.77071931195802
```

As for the SRE, HadaMAG.jl provides several backends for mana computation, including

:serial, :threads, and :mpi_threads; see the documentation for details. Using the MPI backend, we compute the mana of a pure state of $N = 14$ qutrits in roughly 2 hours on four computing nodes, as reported in Table II.

We then evaluate the reduced density matrix ρ for a subsystem of $N_A < N$ qutrits by tracing out the degrees of freedom outside A, and compute the mana $\mathcal{M}(\rho)$ of the resulting mixed state using Algorithm 6.

```julia
julia> using HadaMAG

julia> psi = rand_haar(10; depth=3, q=3)
StateVec{ComplexF64,3}(n=10, dim=59049, mem=923.0 KiB)

julia> rho=reduced_density_matrix(psi, 8)
DensityMatrix{ComplexF64,3}(n=8, dim=6561, mem=657.0 MiB)

julia> Mana(rho)
5.138534117189299
```

The mixed-state algorithm, Algorithm 6, differs from the SRE and pure-state mana routines in that it does not involve a loop over X-strings, making its parallelization more involved. At present, HadaMAG.jl provides only the :serial backend for mixed-state mana, enabling computations on a single CPU core. Nevertheless, thanks to the underlying fast-transform scheme, Algorithm 6 allows us to compute the mana of reduced density matrices ρ for subsystems up to $N_A = 10$ qutrits within a few minutes. The main limitation for larger N_A is not the runtime but the memory required to store the full density matrix ρ, whose dimension grows as $3^{N_A} \times 3^{N_A}$.

	N	n_{nodes}	t_w	$t_{\mathrm{CPU}}^{\mathrm{tot}}$	RAM [MB]		N_A	t_w	RAM [MB]
pure states	6	4	14 ms	6.5 s	229	mixed-states	2	< 1 ms	< 1
	8	4	18 ms	8.1 s	344		4	< 1 ms	< 1
	10	4	0.12 s	54.8 s	573		6	29 ms	9
	12	4	1.28 min	9.58 h	2408		8	3.06 s	706
	14	4	2.29 h	1026 h	18579		10	5.60 min	57128

TABLE II. Performance of mana computation in HadaMAG.jl. **Left:** Mana computation with Algorithm 5 for an N-qutrit random state using 4 nodes, with one core per MPI task and 112 CPU cores per node. Here t_w is the wall-clock time, $t_{\mathrm{CPU}}^{\mathrm{tot}}$ is the total CPU time ($t_{\mathrm{CPU}}^{\mathrm{tot}} = t_w \times 4 \times 112$), and RAM is the total memory usage in megabytes. **Right:** Runtime and memory usage for mixed-state mana calculations with Algorithm 6 on a subsystem of size N_A using a single CPU core. Here t_w is the wall-clock time and RAM is the peak memory consumption in megabytes. The benchmarks were performed on the same hardware as in Table. I.

V. CONCLUSION

A. Summary

In this work we introduced a family of algorithms for the efficient evaluation of SRE and mana, applicable to both pure and mixed many-body states. At the core of our approach are the numer-

ically exact Algorithm 2 for SRE and Algorithm 5 for mana, whose computational cost scales as $O(Nd^{2N})$ for local Hilbert space dimension d. Our algorithms provide an exponential improvement over the naive schemes (Algorithms 1 and 4), which scale as $O(d^{3N})$, and achieve this speedup without introducing any additional memory overhead: the memory footprint remains $O(d^N)$, set by the state vector representation of the analyzed quantum states and a small number of work arrays of the same length.

From a practical standpoint, the introduced algorithms exhibit a high degree of parallelism and are well suited for modern high-performance computing architectures. All core routines are implemented in the open-source HadaMAG.jl package, which leverages multi-threading, MPI-based distributed memory parallelism, and GPU acceleration via CUDA. This enables numerically exact calculations of SRE and mana for system sizes that are significantly beyond the reach of naive implementations, and makes resource-theoretic diagnostics readily available for large-scale numerical studies of quantum dynamics.

On the conceptual side, the fast Hadamard transform (for SRE) and its ternary analogue (for mana) reveal a unifying mathematical structure underlying seemingly different non-stabilizerness measures. In the qubit case, the efficient SRE algorithm recasts the full set of Pauli overlaps into a sequence of Walsh–Hadamard transforms on $\mathbb{Z}_2^N$, while in the qutrit case the efficient mana algorithm expresses Wigner function evaluations as fast Fourier transforms on $\mathbb{Z}_3^N$. This common "fast transform" viewpoint elucidates why both quantities can be computed in $O(Nd^{2N})$ time and suggests possibilities of generalizations to higher local dimensions and other non-stabilizerness measures, such as the generalized stabilizer entropies [46].

The same fast-transform framework also underpins our sampling scheme for SRE, Algorithm 3, which combines thermodynamic integration with a fast-Hadamard-based evaluation of effective energies. By sampling only over X-patterns and using the fast transform to collapse the sum over Z-patterns, the algorithm avoids the exponential growth in the required number of samples with system size that plagues naive Monte Carlo schemes. Our results for quantum circuit states suggest that for generic states the statistical uncertainty on the SRE M_2 grows at most polynomially with N, so that a fixed target precision can be maintained with a polynomial increase of sampling effort, rather than an exponential one.

For mixed qutrit states, we further showed that the fast-transform strategy extends to mana, leading to the mixed-state algorithm Algorithm 6. Here the role of the simple Hadamard transform is played by a structured linear map $M^{\otimes N}$ that connects the vectorized density matrix to the vector of phase-space expectation values. Although the single-qutrit map M has a more intricate internal structure than a Fourier matrix, it can still be applied in an in-place, tensor-factorized fashion, yielding an overall complexity $O(N9^N) = O(Nd^{2N})$ and an exponential speedup over the naive $O(d^{3N})$ approach based on explicit evaluation of expectation values of all phase-space point operators.

B.   Discussion and outlook

Our algorithms are close to optimal from a computational-complexity perspective. Both for qubits (SRE) and for qutrits (mana), the efficient pure-state algorithms evaluate all d^{2N} expectation values of Pauli strings or phase-space point operators, respectively. Given the overall cost $O(Nd^{2N})$, this implies that a *single* expectation value is obtained at a cost that scales only linearly with the system size N. Among methods that explicitly compute the full generalized Pauli spectrum [45] (i.e. all d^{2N} expectation values of generalized Pauli strings), our algorithms are therefore optimal up to a benign linear factor in N. Moreover, they operate at a memory cost proportional to that needed to store

the state vector of the underlying quantum state, without introducing any significant additional memory overheads.

An interesting open question is whether one can bypass the explicit evaluation of the full Pauli (or phase-space) spectrum and still obtain numerically exact values of non-stabilizerness measures such as M_2 or mana, with strictly lower asymptotic cost than $O(Nd^{2N})$. Our results do not rule out such "compressed" schemes, but they set a natural benchmark for any algorithm that proceeds via expectation values. A similar near-optimal behavior appears in the sampling setting: the thermodynamic-integration scheme for SRE achieves an effective linear in N computational cost per sampled Pauli string, and the mixed-state mana algorithm requires only a linear-in-N overhead to evaluate a single phase-space expectation value. In this sense, both the exact and sampling-based approaches presented here saturate, up to polynomial factors, the natural scaling dictated by the number of independent expectation values that must be accessed to evaluate SRE and mana.

It is instructive to compare our approach with existing methods for estimating non-stabilizerness. The sampling algorithm of Ref. [115] provides an unbiased estimator of SRE by generating Pauli samples at a cost scaling as $O(8^{N/2})$ per sample, without incurring any significant memory overhead. However, in that scheme the variance of the Monte Carlo estimator for the SRE M_q at $q > 1$ can grow exponentially with system size, so that maintaining a fixed statistical precision still requires an exponential increase of the number of samples with N. Another important class of approaches relies on tensor-network representations of quantum states. In Ref. [33], SRE is computed directly from a matrix product state (MPS) approximation, with a cost scaling as χ^{6q} for integer Rényi index q, where χ is the MPS bond dimension. Ref. [100] develops an MPS-based sampler that draws *independent* samples from the distribution $\pi(P)$ in Eq. (14) at cost $O(N\chi^3)$ per sample. In Ref. [116], SRE is evaluated via matrix-product operator/ MPS contractions, again with a cost, after MPS compression, proprtional to χ^4. All these tensor-network-based methods are extremely powerful whenever the entanglement of the state is modest, so that a relatively small bond dimension χ provides an accurate approximation and the cost remains polynomial in N.

In contrast to tensor-network approaches, the algorithms introduced in this work operate directly on state-vector representations and make no assumption of low entanglement, complementing the tensor network approaches. The HadaMAG.jl package is therefore particularly well suited for scenarios in which many-body wave functions are explicitly available as state vectors and computing measures of non-stabilizerness via our exact state-vector algorithms is the optimal strategy. Prominent applications include non-equilibrium dynamics of quantum circuits, where magic growth and spreading have been analyzed in detail (e.g. [46, 49, 117]), as well as unitary dynamics of interacting many-body systems governed by ergodic Hamiltonians or Floquet operators [47, 118, 119]. Related settings encompass Mpemba effects [120] and local spreading of magic [117]. In these setups, the time-evolved states for comparatively large systems can be generated efficiently by exploiting the sparse structure of the Hamiltonian [121]. Similarly, methods utilizing sparse Hamiltonian structure [91] provide access to for highly excited eigenstates of many-body systems. The same tools are directly applicable to non-ergodic phases such as many-body localized systems [90], where SRE and mana have already been used to characterize eigenstate properties and dynamics [122]. Other directions for applications involve constrained dynamics and weak ergodicity breaking, including quantum many-body scars [123, 124], as well as lattice gauge theories [38]. Beyond these, HadaMAG.jl can be employed to probe magic in highly excited mid-spectrum states [45] or in ground and excited states of models with non-local interactions, such as Sachdev–Ye–Kitaev-type systems and their variants (e.g. [39, 40, 52]). Finally, the algorithms are also well adapted to monitored and hybrid unitary-measurement dynamics, where entanglement and magic exhibit rich non-equilibrium phenomenology (e.g. [125–130]).

Several directions for future work naturally emerge from our results. On the conceptual side,

the fast-transform perspective developed here may extend to other magic monotones and witnesses beyond SRE and mana, including quantities based on alternative operator bases (e.g. fermionic Gaussians, stabilizer decompositions with restricted Clifford structure, e.g. by symmetry [131]). In parallel, another open question is whether the fast-transform to evaluation of non-stabilizerness measures can be utilized in experiments, e.g., by integrating it with classical post-processing of randomized measurements or shadow tomography.

Finally, the combination of exact and sampling-based tools developed here opens the way to systematic, large-scale studies of magic dynamics and resource distribution across a broad range of many-body platforms: noisy quantum circuits, monitored and hybrid unitary–measurement systems, Floquet and Hamiltonian dynamics in ergodic and non-ergodic phases, as well as models with long-range or all-to-all interactions such as SYK-like systems. In all these settings, non-stabilizerness is expected to play a central role in both enabling and constraining quantum advantage. We therefore anticipate that the algorithms and software introduced in this work will provide a useful baseline for future numerical investigations of non-stabilizerness, particularly in strongly entangled regimes where state-vector methods are the tools of choice.

VI. ACKNOWLEDGMENTS

Note Added: While finalizing this manuscript, we became aware of two independent works Refs. [132, 133], which employ fast Hadamard transform to compute stabilizer Rényi entropies. In parallel, while the HadaMAG.jl codes were developed and optimized, one of the authors (P.S.) applied to the methods of the present manuscript in investigations of non-stabilizerness in quantum circuits [46, 117, 120], ergodic [47], and non-ergodic [122] quantum many-body systems.

[1] J. Preskill, Quantum computing and the entanglement frontier (2012), arXiv:1203.5813 [quant-ph].
[2] A. J. Daley, I. Bloch, C. Kokail, S. Flannigan, N. Pearson, M. Troyer, and P. Zoller, Nature **607**, 667 (2022).
[3] G. Vidal, Phys. Rev. Lett. **91**, 147902 (2003).
[4] R. Orús, Ann. Phys. **349**, 117 (2014).
[5] R. Orús, Nat. Rev. Phys. **1**, 538–550 (2019).
[6] S.-J. Ran, E. Tirrito, C. Peng, X. Chen, L. Tagliacozzo, G. Su, and M. Lewenstein, *Tensor network contractions: methods and applications to quantum many-body systems* (2020).
[7] M. C. Bañuls, Annu. Rev. Condens. Matter Phys. **14**, 173–191 (2023).
[8] F. Verstraete, V. Murg, and J. Cirac, Adv. Phys. **57**, 143 (2008).
[9] N. Schuch, M. M. Wolf, F. Verstraete, and J. I. Cirac, Phys. Rev. Lett. **100**, 030504 (2008).
[10] U. Schollwöck, Ann. Phys. **326**, 96 (2011).
[11] L. G. Valiant, in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01 (Association for Computing Machinery, New York, NY, USA, 2001) p. 114–123.
[12] B. M. Terhal and D. P. DiVincenzo, Phys. Rev. A **65**, 032325 (2002).
[13] S. Bravyi, Quantum Info. Comput. **5**, 216–238 (2005).

[14] D. Gottesman, The heisenberg representation of quantum computers (1998), arXiv:quant-ph/9807006 [quant-ph].

[15] S. Aaronson and D. Gottesman, Phys. Rev. A **70**, 052328 (2004).

[16] P. Sierant, P. Stornati, and X. Turkeshi, Fermionic magic resources of quantum many-body systems (2025), arXiv:2506.00116 [quant-ph].

[17] Z.-W. Liu and A. Winter, PRX Quantum **3**, 020333 (2022).

[18] V. Veitch, S. A. H. Mousavian, D. Gottesman, and J. Emerson, New J. Phys. **16**, 013009 (2014).

[19] A. Heimendahl, M. Heinrich, and D. Gross, Journal of Mathematical Physics **63**, 112201 (2022), https://pubs.aip.org/aip/jmp/article-pdf/doi/10.1063/5.0085774/16591566/112201_1_online.pdf.

[20] T. Haug and L. Piroli, Quantum **7**, 1092 (2023).

[21] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (2000).

[22] S. Bravyi and A. Kitaev, Phys. Rev. A **71**, 022316 (2005).

[23] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Phys. Rev. A **52**, 3457 (1995).

[24] S. Bravyi, G. Smith, and J. A. Smolin, Phys. Rev. X **6**, 021043 (2016).

[25] S. Bravyi and D. Gosset, Phys. Rev. Lett. **116**, 250501 (2016).

[26] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, Quantum **3**, 181 (2019).

[27] M. Howard and E. Campbell, Phys. Rev. Lett. **118**, 090501 (2017).

[28] M. Heinrich and D. Gross, Quantum **3**, 132 (2019).

[29] S. Sarkar, C. Mukhopadhyay, and A. Bayat, New Journal of Physics **22**, 083077 (2020).

[30] H. Hamaguchi, K. Hamada, and N. Yoshioka, Quantum **8**, 1461 (2024).

[31] L. Leone, S. F. E. Oliviero, and A. Hamma, Phys. Rev. Lett. **128**, 050402 (2022).

[32] L. Leone and L. Bittel, Phys. Rev. A **110**, L040403 (2024).

[33] T. Haug and L. Piroli, Phys. Rev. B **107**, 035148 (2023).

[34] P. S. Tarabunga, E. Tirrito, T. Chanda, and M. Dalmonte, PRX Quantum **4**, 040317 (2023).

[35] J. Odavić, T. Haug, G. Torre, A. Hamma, F. Franchini, and S. M. Giampaolo, SciPost Phys. **15**, 131 (2023).

[36] G. Passarelli, R. Fazio, and P. Lucignano, Phys. Rev. A **110**, 022436 (2024).

[37] P. S. Tarabunga and C. Castelnovo, Quantum **8**, 1347 (2024).

[38] P. R. N. Falcão, P. S. Tarabunga, M. Frau, E. Tirrito, J. Zakrzewski, and M. Dalmonte, Phys. Rev. B **111**, L081102 (2025).

[39] B. Jasser, J. Odavic, and A. Hamma, Stabilizer entropy and entanglement complexity in the sachdev-ye-kitaev model (2025), arXiv:2502.03093 [quant-ph].

[40] S. Bera and M. Schirò, Non-stabilizerness of sachdev-ye-kitaev model (2025), arXiv:2502.01582 [quant-ph].

[41] Y.-M. Ding, Z. Wang, and Z. Yan, Evaluating many-body stabilizer rényi entropy by sampling reduced pauli strings: singularities, volume law, and nonlocal magic (2025), arXiv:2501.12146 [quant-ph].

[42] M. Hoshino, M. Oshikawa, and Y. Ashida, Stabilizer rényi entropy and conformal field theory (2025), arXiv:2503.13599 [quant-ph].

[43] M. Hoshino and Y. Ashida, Stabilizer rényi entropy encodes fusion rules of topological defects and boundaries (2025), arXiv:2507.10656 [quant-ph].

[44] D. Rattacaso, L. Leone, S. F. E. Oliviero, and A. Hamma, Phys. Rev. A **108**, 042407 (2023).

[45] X. Turkeshi, A. Dymarsky, and P. Sierant, Phys. Rev. B **111**, 054301 (2025).

[46] X. Turkeshi, E. Tirrito, and P. Sierant, Nature Communications **16**, 2575 (2025).

[47] E. Tirrito, X. Turkeshi, and P. Sierant, Phys. Rev. Lett. **135**, 220401 (2025).

[48] J. Odavić, M. Viscardi, and A. Hamma, Stabilizer entropy in non-integrable quantum evolutions (2025), arXiv:2412.10228 [quant-ph].

[49] T. Haug, L. Aolita, and M. Kim, Quantum **9**, 1801 (2025).

[50] J. A. Montañà López and P. Kos, J. Phys. A: Math. and Theor. **57**, 475301 (2024).

[51] P. S. Tarabunga and E. Tirrito, Magic transition in measurement-only circuits (2024), arXiv:2407.15939 [quant-ph].

[52] G. C. Santra, A. Windey, S. Bandyopadhyay, A. Legramandi, and P. Hauke, Complexity transitions in chaotic quantum systems (2025), arXiv:2505.09707 [quant-ph].

[53] G. Passarelli, A. Russomanno, and P. Lucignano, Nonstabilizerness of a boundary time crystal (2025), arXiv:2503.05243 [quant-ph].

[54] Z.-Y. Hou, C. Cao, and Z.-C. Yang, Stabilizer entanglement as a magic highway (2025), arXiv:2503.20873 [quant-ph].

[55] Y. Zhang and Y. Gu, Quantum magic dynamics in random circuits (2024), arXiv:2410.21128 [quant-ph].

[56] B. Magni, A. Christopoulos, A. D. Luca, and X. Turkeshi, Anticoncentration in clifford circuits and beyond: From random tensor networks to pseudo-magic states (2025), arXiv:2502.20455 [quant-ph].

[57] B. Magni and X. Turkeshi, Quantum complexity and chaos in many-qudit doped clifford circuits (2025), arXiv:2506.02127 [quant-ph].

[58] Y. Zhang, S. Vijay, Y. Gu, and Y. Bao, Designs from magic-augmented clifford circuits (2025), arXiv:2507.02828 [quant-ph].

[59] D. Szombathy, A. Valli, C. P. Moca, L. Farkas, and G. Zaránd, Independent stabilizer rényi entropy and entanglement fluctuations in random unitary circuits (2025), arXiv:2501.11489 [quant-ph].

[60] S. F. E. Oliviero, L. Leone, A. Hamma, and S. Lloyd, npj Quantum Info. **8**, 148 (2022).

[61] T. Haug and M. Kim, PRX Quantum **4**, 010301 (2023).

[62] T. Haug, S. Lee, and M. S. Kim, Phys. Rev. Lett. **132**, 240602 (2024).

[63] R. Sarkar and T. J. Yoder, Quantum **8**, 1307 (2024).

[64] D. Gross, J. Math. Phys. **47**, 122107 (2006).

[65] D. Gross, Applied Physics B **86**, 367 (2007).

[66] V. Veitch, C. Ferrie, D. Gross, and J. Emerson, New Journal of Physics **14**, 113011 (2012).

[67] H. Pashayan, J. J. Wallman, and S. D. Bartlett, Phys. Rev. Lett. **115**, 070501 (2015).

[68] C. D. White, C. Cao, and B. Swingle, Phys. Rev. B **103**, 075145 (2021).

[69] P. S. Tarabunga, Quantum **8**, 1413 (2024).

[70] C. D. White and J. H. Wilson, Mana in haar-random states (2020), arXiv:2011.13937 [quant-ph].

[71] K. Goto, T. Nosaka, and M. Nozaki, Phys. Rev. D **106**, 126009 (2022).

[72] T. J. Sewell and C. D. White, Phys. Rev. B **106**, 125130 (2022).

[73] N. Koukoulekidis and D. Jennings, npj Quantum Information **8**, 42 (2022).

[74] R. Basu, A. Ganguly, S. Nath, and O. Parrikar, Journal of High Energy Physics **2024**, 264 (2024).

[75] A. Ahmadi and E. Greplova, SciPost Phys. **16**, 043 (2024).

[76] R. Nyström, N. Pranzini, and E. Keski-Vakkuri, Phys. Rev. Res. **7**, 033085 (2025).

[77] C. E. P. Robin and M. J. Savage, The magic in nuclear and hypernuclear forces (2024), arXiv:2405.10268 [nucl-th].

[78] F. Brökemeier, S. M. Hengstenberg, J. W. T. Keeble, C. E. P. Robin, F. Rocco, and M. J. Savage, Quantum magic and multi-partite entanglement in the structure of nuclei (2024), arXiv:2409.12064 [nucl-th].

[79] C. D. White and M. J. White, Phys. Rev. D **110**, 116016 (2024).

[80] I. Chernyshev, C. E. P. Robin, and M. J. Savage, Phys. Rev. Res. **7**, 023228 (2025).

[81] Q. Liu, I. Low, and Z. Yin, Quantum magic in quantum electrodynamics (2025), arXiv:2503.03098 [quant-ph].

[82] A. Gu, H.-Y. Hu, D. Luo, T. L. Patti, N. C. Rubin, and S. F. Yelin, Quantum **8**, 1422 (2024).

[83] C. Cao, G. Cheng, A. Hamma, L. Leone, W. Munizzi, and S. F. E. Oliviero, Gravitational back-reaction is magical (2025), arXiv:2403.07056 [hep-th].

[84] F. B. Trigueros, Stabilizer renyi entropies, https://github.com/FabianBallar7/StabilizerRenyiEntropy (2025).

[85] F. B. Trigueros and J. A. M. Guzmán, Nonstabilizerness and error resilience in noisy quantum circuits (2025), arXiv:2506.18976 [quant-ph].

[86] G. Lami and M. Collura, Phys. Rev. Lett. **131**, 180401 (2023).

[87] P. S. Tarabunga, E. Tirrito, M. C. Bañuls, and M. Dalmonte, Phys. Rev. Lett. **133**, 010601 (2024).

[88] L.-Y.-N. Liu, S. Yi, and J. Cui, Stabilizer rényi entropy for translation-invariant matrix product states (2025), arXiv:2508.03534 [quant-ph].

[89] D. J. Luitz and Y. Bar Lev, Phys. Rev. B **96**, 020406 (2017).

[90] P. Sierant, M. Lewenstein, A. Scardicchio, L. Vidmar, and J. Zakrzewski, Reports on Progress in Physics **88**, 026502 (2025).

[91] P. Sierant, M. Lewenstein, and J. Zakrzewski, Phys. Rev. Lett. **125**, 156601 (2020).

[92] A. Morningstar, L. Colmenarez, V. Khemani, D. J. Luitz, and D. A. Huse, Phys. Rev. B **105**, 174205 (2022).

[93] D. J. Luitz and Y. Bar Lev, Phys. Rev. A **99**, 010105 (2019).

[94] N. Defenu, A. Lerose, and S. Pappalardi, Physics Reports **1074**, 1 (2024), out-of-equilibrium dynamics of quantum many-body systems with long-range interactions.

[95] M. Brenes, V. K. Varma, A. Scardicchio, and I. Girotto, Computer Physics Communications **235**, 477 (2019).

[96] Fino and Algazi, IEEE Transactions on Computers **C-25**, 1142 (1976).

[97] L. Bittel and L. Leone, Operational interpretation of the stabilizer entropy (2025), arXiv:2507.22883 [quant-ph].

[98] The construction below applies to any odd prime dimension d.

[99] D. E. Knuth, *The art of computer programming*, Vol. 3 (Pearson Education, 1997).

[100] G. Lami and M. Collura, Phys. Rev. Lett. **131**, 180401 (2023).

[101] P. S. Tarabunga and T. Haug, SciPost Phys. **19**, 085 (2025).

[102] For $q = 1$, the variance is instead at most polynomial in N, enabling an efficient estimation of $M_1(|\psi\rangle)$.

[103] J. de Boer, J. Järvelä, and E. Keski-Vakkuri, Phys. Rev. D **99**, 066012 (2019).

[104] H. Yao and X.-L. Qi, Phys. Rev. Lett. **105**, 080501 (2010).

[105] J. Schliemann, Phys. Rev. B **83**, 115322 (2011).

[106] M. P. Zaletel, J. H. Bardarson, and J. E. Moore, Phys. Rev. Lett. **107**, 020402 (2011).

[107] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[108] W. K. Hastings, Biometrika **57**, 97 (1970).

[109] T. Haug and P. S. Tarabunga, Efficient witnessing and testing of magic in mixed quantum states (2025), arXiv:2504.18098 [quant-ph].

[110] L. Hantzko, L. Binkowski, and S. Gupta, Physica Scripta **99**, 085128 (2024).

[111] T. Jones, Decomposing dense matrices into dense Pauli tensors, arXiv preprint (2024), arXiv:2401.16378 [quant-ph].

[112] S. Vidal Romero and J. Santos-Suárez, Quantum Information Processing **22**, 449 (2023).

[113] X. Turkeshi, M. Schirò, and P. Sierant, Phys. Rev. A **108**, 042408 (2023).

[114] We exclude the identity Pauli string $P = \mathbb{I}$ from the sampling procedure and account explicitly for its contribution, equal to unity for any $|\psi\rangle$, in the post-processing of the results, cf. [45].

[115] P. S. Tarabunga and T. Haug, Efficient mutual magic and magic capacity with matrix product states (2025), arXiv:2504.07230 [quant-ph].

[116] P. S. Tarabunga, E. Tirrito, M. C. Bañuls, and M. Dalmonte, Phys. Rev. Lett. **133**, 010601 (2024).

[117] S. Aditya, X. Turkeshi, and P. Sierant, Growth and spreading of quantum resources under random circuit dynamics (2025), arXiv:2512.14827 [quant-ph].

[118] E. Tirrito, P. S. Tarabunga, D. S. Bhakuni, M. Dalmonte, P. Sierant, and X. Turkeshi, Universal spreading of nonstabilizerness and quantum transport (2025), arXiv:2506.12133 [quant-ph].

[119] J. Odavić, M. Viscardi, and A. Hamma, Phys. Rev. B **112**, 104301 (2025).

[120] S. Aditya, A. Summer, P. Sierant, and X. Turkeshi, Mpemba effects in quantum complexity (2025), arXiv:2509.22176 [quant-ph].

[121] H. Tal-Ezer and R. Kosloff, J. Chem. Phys. **81**, 3967 (1984).

[122] P. R. N. Falcão, P. Sierant, J. Zakrzewski, and E. Tirrito, Phys. Rev. Lett. **135**, 240404 (2025).

[123] M. Serbyn, D. A. Abanin, and Z. Papić, Nature Physics **17**, 675 (2021).

[124] R. Smith, Z. Papić, and A. Hallam, Phys. Rev. B **111**, 245148 (2025).

[125] G. E. Fux, E. Tirrito, M. Dalmonte, and R. Fazio, Phys. Rev. Res. **6**, L042030 (2024).

[126] M. Bejan, C. McLauchlan, and B. Béri, PRX Quantum **5**, 030332 (2024).

[127] P. S. Tarabunga and E. Tirrito, npj Quantum Information **11**, 166 (2025).

[128] A. Paviglianiti, G. Lami, M. Collura, and A. Silva, PRX Quantum **6**, 030320 (2025).

[129] A. Scocco, W.-K. Mok, L. Aolita, M. Collura, and T. Haug, Rise and fall of nonstabilizerness via random measurements (2025), arXiv:2507.11619 [quant-ph].

[130] H. Lóio, G. Lami, L. Leone, M. McGinley, X. Turkeshi, and J. D. Nardis, Quantum state designs via magic teleportation (2025), arXiv:2510.13950 [quant-ph].

[131] S. Cepollaro, G. Cuffaro, M. B. Weiss, S. Cusumano, A. Hamma, and S. Lloyd, Stabilizer entropy of subspaces (2025), arXiv:2512.23013 [quant-ph].

[132] Z. Xiao and S. Ryu, Exponentially accelerated sampling of pauli strings for nonstabilizerness (2026), arXiv:2601.00761 [quant-ph].

[133] X. Huang, H.-Z. Li, and J.-X. Zhong, A fast and exact approach for stabilizer rényi entropy via the xor-fwht algorithm (2026), arXiv:2512.24685 [quant-ph].