Near-threshold qudit stabilizer codes with efficient encoding circuits for magic-state distillation

Abhi Kumar Sharma and Shayan Srinivasa Garani [*]

Department of Electronic Systems Engineering, Indian Institute of Science, Bengaluru, 560012, India

(Received 19 February 2024; revised 26 April 2024; accepted 24 May 2024; published 18 June 2024)

Magic-state distillation is a protocol to purify noisy quantum states using quantum stabilizer codes. In this paper, we provide conditions for identifying magic states along with analytical expressions for computing the noise threshold. Through quantum entropic results, we prove the convergence of the magic-state distillation procedure. We derive analytical results to show how the noise threshold scales as a function of the minimum distance and the number of physical qudits of the underlying quantum stabilizer code. We distill $|T\rangle$ and $|S\rangle$ states with thresholds 0.498 (theoretical limit is 0.5) and 0.71 (theoretical limit is 0.75) over the depolarizing channel using $[[31, 1, 16]]_2$ quantum Bose-Chaudhuri-Hocquenghem code and $[[29, 1, 15]]_3$ quantum qutrit code, respectively. Finally, encoding algorithms based on quantum Calderbank-Shor-Steane (CSS) and non-CSS qudit stabilizer codes along with circuits are proposed for magic-state distillation.

DOI: 10.1103/PhysRevA.109.062426

I. INTRODUCTION

Quantum error correction (QECC) [1–3] is the foundation for fault-tolerant quantum computing. To harness the power of quantum advantage, the realization of fault-tolerant Clifford and non-Clifford gates is essential. Such non-Clifford gates also find applications within quantum machine-learning architectures [4,5]. The implementation of fault-tolerant non-Clifford gates is feasible using *magic* states [6]. Therefore, magic-state distillation (MSD) using QECCs from imperfectly prepared noisy states is of significant importance.

The MSD protocol was originally introduced [6] for the distillation of the eigenstates of the T gate using a five-qubit stabilizer code. Magic states are useful in implementing non-Clifford or rotational gates via state injection. Higher-dimensional codes, such as those from qutrits, with better error suppression compared to the qubit counterparts, were recently proposed by Prakash [7]. The basic set of gates, such as the Hadamard gate H, the S gate, and the controlled-NOT gate [8,9], can generate the complete Clifford group. The eigenstates of the operator from the Clifford group can be used as magic states for implementing non-Clifford operators towards universal fault-tolerant quantum computing.

If the discrete Wigner function over a quantum state evaluates to a negative value, magic states can be distilled [10] from such a quantum state. However, not all stabilizer codes can generate a particular distillable magic state. This opens up the question regarding the choice of stabilizer codes for the distillation of specific magic states, which we would like to address in this paper.

Consider a $[[n, k, d]]_p$ qudit quantum stabilizer code Q that encodes k qudits into n qudits, having minimum distance d. To distill a state from Q using MSD from n noisy qudits [6], stabilizer generators of the code Q are measured such that all generators give $+1$ measurement outcome; if not, the process is repeated. Analytically, measurement is performed using a projection operator constructed from the normalized addition of the stabilizer group elements, where the stabilizer group is generated using the stabilizer generators of Q. The purified magic state is obtained by decoding the projected state to a single-qudit quantum state.

Designing Clifford operators for decoding quantum codes [11,12] can be challenging since the quantum circuit parameters have to be optimized based on the circuit depth, gate complexity, and transversality requirements. Since the depth of a quantum decoding circuit scales with the code length [12], it is desired to maintain lower depth suitable for the decoding step of the MSD protocol. The inverse of the decoding operator can be used for quantum encoding [13].

In this article, we formulate the MSD protocol for general qudit stabilizer codes and provide insights into the working of the procedure from an information-theoretic perspective. We also construct practical codes along with efficient encoding circuits, practically useful for MSD. Our main contributions are summarized as follows: (1) We provide the necessary condition to determine whether a quantum state is distillable using a particular stabilizer code, useful for identifying suitable quantum codes. We provide an explicit formula for computing the noise threshold for the MSD process based on the qudit stabilizer code parameters. (2) By projecting the n-qudit noisy state onto the code space during the purification process, we establish that this syndrome probability is a monotonically nondecreasing function over iterations. Further, we prove that the convergence of a noisy state to a pure state is convergent in a quantum entropic sense, building over the simulations studies carried out in [10]. (3) We provide an analysis to show how the minimum distance and code length influence the noise threshold. We validate the theory by providing $[[31, 1, 16]]_2$ code for the distillation of eigenstate $|T\rangle$ of T gate [6] with the noise threshold of 0.498. Also, we provide $[[13, 1, 7]]_3$ and $[[29, 1, 15]]_3$ codes for the distillation of the eigenstate

[*] Contact author: shayangs@iisc.ac.in

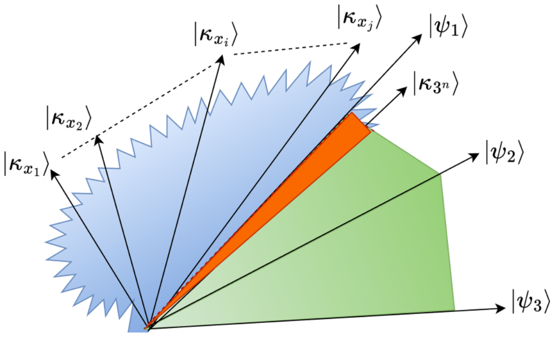


FIG. 1. For a qutrit code, the green simplex spanned by $\{|\phi_1\rangle, |\phi_2\rangle, |\phi_3\rangle\}$ and the blue simplex spanned by $\{|\xi_0\rangle, |\xi_1\rangle, \dots, |\xi_{3^n-1}\rangle\}$ represent the single-qutrit code space and the 3^n-dimensional qutrit state input to the MSD, respectively. The two simplexes intersect within the orange region in the middle where distillable states lie.

$|S\rangle$ of the qutrit Hadamard gate with the noise thresholds of 0.425 and 0.71, where the former code can be used when the quantum hardware has a limited number of noisy operational qutrits, and the latter code can be used to fulfill the requirements of higher noise thresholds for noisy quantum hardware with sufficient number of qutrits, thereby proving that codes with larger minimum distance improves the noise threshold while distilling the magic states. (4) Finally, we provide algorithms to design encoding circuits for CSS as well as hypergraph product codes, practically useful for building qudit stabilizer codes for MSD.

The paper is organized as follows: In Sec. II, we begin with a discussion on the geometric framework of the MSD algorithm based on qudit stabilizer codes. We provide the conditions for distillation and computing noise threshold. In Sec. III, we provide analytical insights into the convergence of the MSD algorithm based on quantum entropic changes to the noisy state over iterations and the scaling of probability of error with respect to the minimum distance of the code. In Sec. IV, we provide practical codes to distill the $|T\rangle$ and $|S\rangle$ states [7]. We also prove that the noise threshold improves with the minimum distance of the quantum code. In Sec. V, we provide algorithms to encode any Calderbank-Shor-Steane (CSS) and non-CSS qudit stabilizer codes. The latter uses the hypergraph product code construction to create a qudit quantum stabilizer code. This is useful to illustrate the role of quantum decoding circuits based on qudit stabilizer codes for MSD. We conclude the paper in Sec. VI.

II. MAGIC-STATE DISTILLATION USING QUDITS

A quantum stabilizer code with a high noise threshold for the MSD is generally challenging to search. In a geometric sense, the choice of a stabilizer code for a given noisy magic quantum state depends upon the alignment of the noisy state with the logical states spanning the subspace of the stabilizer code. This is illustrated in Fig. 1 for the qutrit case. A single-dimensional code is typically used [6,7,10] in MSD to avoid any degeneracy over multiple logical states of the quantum stabilizer code. For a p-level qudit, a general stabilizer code $[[n, 1, d]]_p$, with associated generators $\langle s_1, s_2, \dots, s_{n-1}\rangle$, has a qudit projection operator [10] given by

$$\mathrm{P} = \frac{1}{p^{n-1}} \prod_{k=1}^{n-1} \sum_{l=0}^{p-1} s_k^l \, . \tag{1}$$

If all the generators $s_i, \ \forall\, i$, give a $+1$ measurement outcome on measuring an n-qudit state, then the operator P will project the state onto the code space spanned by the eigenvectors of P. The spectral decomposition of the operator P can bring more insights into the MSD protocol as follows:

$$\mathrm{P} = \sum_{k=1}^{p} \alpha_k |\phi_k\rangle\langle\phi_k|, \tag{2}$$

where $\alpha_k = 1$ for a projection operator since $|\phi_j\rangle$'s are the basis of the code space; therefore, all elements of the stabilizer group will give a $+1$ eigenvalue for $|\phi_j\rangle$. The projection space spanned by $|\phi_j\rangle, \ \forall\, j$, and the common eigenvectors of s_i, for all i, is the code space [1].

A. Background of MSD

Any quantum state σ over a p-level qudit can be expressed using $p^2 - 1$ Heisenberg-Weyl basis operators [6]. Such $p^2 - 1$ parameters of the state σ can be reduced to $p - 1$ by randomly applying a single-qudit Clifford unitary U. The eigenstates of the operator U are $|\xi_i\rangle$ with eigenvalues α_i. We note that $U^m = \mathrm{I}$ over σ.

One of the eigenstates of the operator U is the actual magic state, which can be distilled through the MSD process, and the remaining eigenstates are just noisy states. To reduce the dimensionality of the state σ, we apply a twirling scheme as follows:

$$T(\sigma) = \frac{1}{m} \sum_{i=0}^{m-1} U^i \sigma (U^\dagger)^i = \sum_{i=0}^{p-1} c_i |\xi_i\rangle\langle\xi_i|, \tag{3}$$

where $w_l = \langle\xi_l|\sigma|\xi_l\rangle$. From Eq. (3), we get

$$w_l = \langle\xi_l| \frac{1}{m} \sum_{i=0}^{m-1} U^i \sigma (U^\dagger)^i |\xi_l\rangle. \tag{4}$$

Using the spectral decomposition of U, we get

$$w_l = \frac{1}{m} \sum_{i=0}^{m-1} \sum_{j,k=0}^{p-1} \alpha_j^i \left(\alpha_k^i\right)^* \langle\xi_j|\sigma|\xi_k\rangle \langle\xi_l|\xi_j\rangle \langle\xi_k|\xi_l\rangle. \tag{5}$$

Since $|\xi_j\rangle$ are orthonormal, we get

$$w_l = s_l \langle\xi_l|\sigma|\xi_l\rangle, \text{ where } s_l = \frac{1}{m} \sum_{i=0}^{m-1} |\alpha_l|^{2i}. \tag{6}$$

As $\sum_{l=0}^{p-1} w_l = \sum_{l=0}^{p-1} \langle\xi_l|\sigma|\xi_l\rangle = 1$. Therefore, m is the minimum value for which $s_l = 1, \ \forall\, l$. As a result,

$$w_l = \langle\xi_l|\sigma|\xi_l\rangle. \tag{7}$$

We can project state σ onto the hyperplane spanned by the eigenstates of the operator U, i.e., span$(|\xi_0\rangle, |\xi_1\rangle, \dots, |\xi_{p-1}\rangle)$. The hyperplane for the state σ can be parametrized and expressed in terms of the eigenstates of U, as follows:

$$T(\sigma) = \left(1 - \sum_{i=1}^{p-1} \epsilon_i\right)|\xi_0\rangle\langle\xi_0| + \sum_{i=1}^{p-1} \epsilon_i |\xi_i\rangle\langle\xi_i|. \tag{8}$$

First step: We prepare a state $\sigma' = T(\sigma)^{\otimes n}$, which is an input to the MSD protocol. The state σ' can be expressed in terms of eigenstates of U using Eq. (8) as follows:

$$\sigma' = \sum_{\underline{z}\in\mathbb{F}_p^n} w_{\underline{z}} |\xi_{\underline{z}}\rangle\langle\xi_{\underline{z}}|, \tag{9}$$

where $|\xi_{\underline{z}}\rangle = |\xi_{z_1}\rangle \otimes \cdots \otimes |\xi_{z_n}\rangle$.

Second step: All the stabilizer generators of the code are measured postselected on the $+1$ measurement[1] outcome. The projected state ρ is as follows:

$$\rho = \frac{\mathrm{P}\sigma'\mathrm{P}^\dagger}{\mathrm{tr}(\sigma'\mathrm{P})}. \tag{10}$$

Third step: A Clifford unitary D is applied on the projected state ρ. Operator D sends the logical operators of the code to a single-qudit operator. A single-qudit state σ_t [6] is obtained by decoding the projected state ρ as follows:

$$D\rho D^\dagger = \sigma_t \otimes |0\rangle\langle 0|^{\otimes(n-1)} \tag{11}$$

where D is a decoding operator. The operator D can transform an n-qudit state to single-qudit state and $n-1$ ancillae in the $|0\rangle^{\otimes(n-1)}$ state.

B. Conditions to identify the distilllable magic states

Consider a $[[n, 1, d]]_p$ quantum stabilizer code with transversal logical operators [6,10,14] having a projection operator P with eigenvectors $|\phi_0\rangle, |\phi_1\rangle, \dots, |\phi_{p-1}\rangle$ and eigenvalue $+1$. All stabilizer generators of the code and their combinations give a measurement outcome of $+1$ if the state belongs to the code space. So, $|\phi_0\rangle, |\phi_1\rangle, \dots, |\phi_{p-1}\rangle$ span the projection space of the quantum stabilizer code. After step 2 of MSD discussed in Sec. II A, the projected state ρ in Eq. (10) becomes

$$\begin{aligned}\rho &= \frac{\sum_{i,j\in\mathbb{F}_p} |\phi_i\rangle\langle\phi_i|\sigma'|\phi_j\rangle\langle\phi_j|}{P_s}\\ &= \frac{1}{P_s}\sum_{i,j\in\mathbb{F}_p} |\phi_i\rangle\langle\phi_i| \left(\sum_{\underline{z}\in\mathbb{F}_p^n} w_{\underline{z}}|\xi_{\underline{z}}\rangle\langle\xi_{\underline{z}}|\right)|\phi_j\rangle\langle\phi_j|\\ &= \frac{1}{P_s}\left[\sum_{i\in\mathbb{F}_p}\sum_{\underline{z}\in\mathbb{F}_p^n} w_{\underline{z}}|\langle\phi_i|\xi_{\underline{z}}\rangle|^2|\phi_i\rangle\langle\phi_i|\right.\\ &\quad\left. + \sum_{(i,j)\in\mathbb{F}_p^2:i\neq j}\sum_{\underline{z}\in\mathbb{F}_p^n} w_{\underline{z}}|\langle\phi_i|\xi_{\underline{z}}\rangle\langle\xi_{\underline{z}}|\phi_j|\rangle|\phi_i\rangle\langle\phi_j|\right],\end{aligned} \tag{12}$$

where $P_s = \sum_{i\in\mathbb{F}_p}\langle\phi_i|\sigma'|\phi_i\rangle$. Replacing the variables $w_{\underline{z}}$ and $\langle\xi_{\underline{z}}||\phi_j\rangle$ with functions p_i and $q_{i,j}$, we get

$$\rho = \frac{\sum_{i\in\mathbb{F}_p} p_i|\phi_i\rangle\langle\phi_i|}{P_s} + \frac{\sum_{i,j\in\mathbb{F}_p:i\neq j} q_{i,j}|\phi_i\rangle\langle\phi_j|}{P_s}, \tag{13}$$

where

$$p_i = \sum_{\underline{z}\in\mathbb{F}_p^n} w_{\underline{z}}|\langle\phi_i||\xi_{\underline{z}}\rangle|^2, \tag{14}$$

$$q_{i,j} = \sum_{\underline{z}\in\mathbb{F}_p^n} w_{\underline{z}}|\langle\phi_i||\xi_{\underline{z}}\rangle\langle\xi_{\underline{z}}||\phi_j\rangle|. \tag{15}$$

Defining set $\mathcal{W} = \{\underline{z} : \mathrm{P}|\xi_{\underline{z}}\rangle = 0\}$, Eqs. (14) and (15) become

$$p_i = \sum_{\underline{z}\in\mathbb{F}_p^n\backslash\mathcal{W}} w_{\underline{z}}|\langle\phi_i||\xi_{\underline{z}}\rangle|^2, \tag{16}$$

$$q_{i,k} = \sum_{\underline{z}\in\mathbb{F}_p^n\backslash\mathcal{W}} w_{\underline{z}}|\langle\phi_i||\xi_{\underline{z}}\rangle\langle\xi_{\underline{z}}||\phi_k\rangle|. \tag{17}$$

Since cross fidelities $q_{i,k} = 0$ for all i and k using the projection equivalence equation[2] [6], Eq. (13) simplifies to

$$\rho = \frac{1}{P_s}\sum_{i\in\mathbb{F}_p} p_i|\phi_i\rangle\langle\phi_i|. \tag{18}$$

At the output of the protocol, the probability $\frac{p_0}{P_s}$ of the state $|\phi_0\rangle$, which is proportional to the state $|\xi_0\rangle$, should be nonzero[3] and greater than the initial syndrome probability of the input magic state $|\xi_0\rangle$.

C. Calculation of the noise threshold

In this section, we analyze the noise threshold for distilling the nonstabilizer state, i.e., the magic state using a p-level qudit quantum code. The noise threshold of a state is a point beyond which the MSD protocol cannot purify the noisy state. From Eq. (16), we infer that the fidelity functions do not alter the decoding operations since it is simply a mapping of the logical operators of the code to single-qudit unitary operators [10].

Consider a single parameter η reduced from the parameters $\omega_1, \dots, \omega_{p-1}$ given by $\eta = \sum_{i=1}^{p-1}\omega_i$. This parameter reduction can be visualized as projecting a state from the hyperplane onto a line with $\omega_i = \frac{\eta}{p-1}$, $\forall\ i$.

Figure 2 shows the geometry of magic-state distillation after twirling, i.e., postdimensionality reduction for a qutrit Hadamard operator. A single parameter η controls the state ρ to lie on a line. When the state lies within the noise threshold, i.e., confined to the distillable regions, we can achieve magic-state distillation.

[1] The generators of the stabilizer code will give $+1$ measurement outcome if the state σ' in the first step is either a codeword or belongs to the code space.

[2] The projection [6] equation is given by

$$\sum_{i=0}^{p-1}|\phi_i\rangle\langle\phi_i| = \sum_{\underline{x}\in\mathbb{F}_p^n}\mathrm{P}|\xi_{\underline{x}}\rangle\langle\xi_{\underline{x}}|\mathrm{P}.$$

[3] The pure magic state prepared n times should not lie outside the code space.

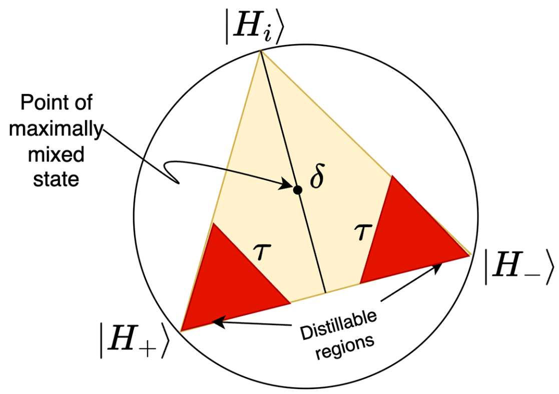


FIG. 2. The vertices of the equilateral triangle represent the eigenstates of the qutrit Hadamard operator. The state ρ lies on this plane after the twirling scheme. The parameter δ varies the state ρ confined to a line. The red regions (distillable regions) contain the distillable states restricted to the noise threshold parameter τ.

We can rewrite Eq. (8) in terms η as

$$T(\rho) = (1-\eta)|\xi_0\rangle\langle\xi_0| + \left(\frac{\eta}{p-1}\right)\sum_{i=1}^{p-1}|\xi_i\rangle\langle\xi_i|, \qquad (19)$$

where $|\xi_0\rangle$ is the distillable state and $|\xi_j\rangle,\ j\in\{1,\dots,p-1\}$, are the noisy states. Following the approach from [6], the density matrix σ' in Eq. (9) can be rewritten as follows:

$$\sigma' = \sum_{\underline{z}\in\mathbb{F}_p^n}(1-\eta)^{n-H(\underline{z})}\left(\frac{\eta}{p-1}\right)^{H(\underline{z})}|\xi_{\underline{z}}\rangle\langle\xi_{\underline{z}}|, \qquad (20)$$

where $H(\underline{z})$ is the Hamming weight of $\underline{z}$. The fidelity functions p_i in Eq. (16) become

$$p_i = \sum_{\underline{z}\in\mathbb{F}_p^n\setminus\mathcal{W}}(1-\eta)^{n-H(\underline{z})}\left(\frac{\eta}{p-1}\right)^{H(\underline{z})}|\langle\phi_i||\xi_{\underline{z}}\rangle|^2, \qquad (21)$$

where $\mathcal{W}$ is the index set for which $\mathrm{P}|\xi_{\underline{z}}\rangle = 0$. The projection probability defined from (18) satisfies the properties of the density matrix.

$$P_s = \mathrm{tr}(\sigma'\mathrm{P}) = \sum_{i\in\mathbb{F}_p} p_i. \qquad (22)$$

To calculate the noise threshold, we equate the probability of error in the input and output states defined in [6]. Therefore, we can extend the threshold calculation for the general qudits by following a similar approach. We calculate $\eta_{\text{th}} = \sum_{i=1}^{p-1}\frac{p_i}{\mathrm{tr}(\sigma'\mathrm{P})}$. Equation (21) is rewritten as

$$\sum_{i\in S\setminus\{0\},\underline{z}\in\mathbb{F}_p^n\setminus W}(1-\eta_{\text{th}})^{n-H(\underline{z})}\left(\frac{\eta_{\text{th}}}{p-1}\right)^{H(\underline{z})}|\langle\phi_i||\xi_{\underline{z}}\rangle|^2$$
$$-\,\mathrm{tr}(\sigma'\mathrm{P})\eta_{\text{th}} = 0, \qquad (23)$$

where $S = \{0,1,\dots,p-1\}$. We can expand $\mathrm{tr}(\sigma'\mathrm{P})$ in terms of p_i in Eq. (23) by using Eq. (22). On further simplification, we can calculate η_{th} in terms of code parameters as follows:

$$\eta_{\text{th}} = \frac{\sum_{i\in S\setminus\{0\}}|\langle\phi_i||\xi_{\underline{0}}\rangle|^2}{\sum_{i\in S}|\langle\phi_i||\xi_{\underline{0}}\rangle|^2}, \qquad (24)$$

where $|\xi_{\underline{0}}\rangle = |\xi_0\rangle^{\otimes n}$.

Example 1. The distillation of the eigenstates of the qutrit Hadamard gate H was proposed using a five-qutrit quantum stabilizer code $Q_1 = [[5,1,3]]_3$ detailed in Table 2 of [10]. $|H_{\pm1}\rangle$ are the distillable states using Q_1, but the state $|H_i\rangle$ is not distillable. Now, we check using our procedure if the state $|H_i\rangle$ is distillable or not using the code Q_1.

Consider the noisy qutrit state σ twirled as follows:

$$\sigma \to \frac{1}{4}\sum_{i=1}^{4}H^i\sigma(H^i)^\dagger. \qquad (25)$$

Post twirling, σ is projected from an eight-dimensional Hilbert space to a two-dimensional space, i.e., on a Hadamard plane as shown in Fig. 2. The distillable states lie in the red region [10] within the threshold. σ can be written as $\sigma = (1-\omega_1-\omega_2)|H_i\rangle\langle H_i| + \omega_1|H_1\rangle\langle H_1| + \omega_2|H_{-1}\rangle\langle H_{-1}|$. With $\delta = \omega_1+\omega_2$ and $\omega_1=\omega_2=\frac{\delta}{2}$, following Eq. (19), σ is confined to a line controlled by δ given by

$$\sigma = (1-\delta)|H_i\rangle\langle H_i| + \frac{\delta}{2}(|H_{-1}\rangle\langle H_{-1}| + |H_1\rangle\langle H_1|). \qquad (26)$$

The fidelity functions $\{p_i\}_{i=0}^2$ for the state $|H_i\rangle$ are $\{0, 0.0144, 0.0145\}$ for $\delta = 0$. Therefore, using the five-qutrit code, the state $|H_i\rangle$ is not distillable. We have simulated the results of Sec. IV in [10] for the states $|H_{\pm1}\rangle$. The $|H_{\pm1}\rangle$ states are distillable when ϵ_1 and ϵ_2 are less than $\eta/2$. The noise threshold evaluated using simulation is around[4] $0.21\times\frac{9}{8}\approx 0.236$, as shown in Fig. 3. We compute the threshold for states $|H_{\pm1}\rangle$ by Eq. (24) to be 0.237.

III. CONVERGENCE ANALYSIS OF THE MSD PROCEDURE

Quantum entropic measure can be used to quantify the mixedness in a quantum state. We can track the MSD process by calculating the quantum entropy of the output state in the subsequent iterations of the MSD process. If we are operating within the threshold region, then every iteration of MSD leads to the convergence of a state in the quantum entropic sense as we approach pure states.

To prove it, we need to establish that the quantum entropy of the output state ρ in (18) is less than the quantum entropy of the input state σ' in (9) over every MSD iteration.

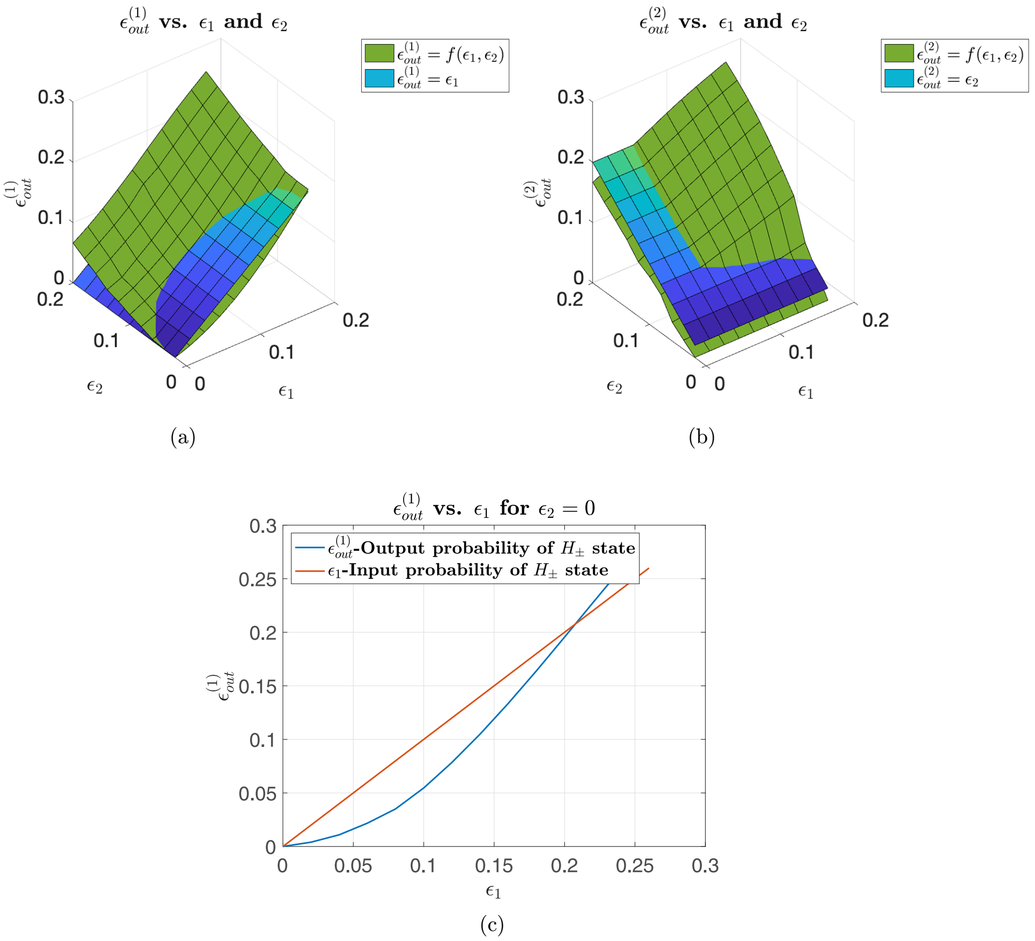
Theorem 1. For a p-level qudit code, the quantum entropies of the input state σ_{in} and the output state σ_{out} at the tth iteration of the MSD protocol are related as follows:

$$H(\sigma)_{\text{out}}^{(t)} \leqslant \frac{1}{P_s}\left(\log_2(P_s) + H(\sigma)_{\text{in}}^{(t)}\right).$$

We prove Theorem 1 in Appendix A.

In Example 1, we discuss the distillation of the eigenstates of the Hadamard gate using $[[5,1,3]]_3$ code. As $|H_i\rangle$ is not

[4] The qutrit state ρ when affected with the depolarizing noise becomes $\rho_1 = (1-p)\rho + \frac{p}{3}I_3$. The Kraus operators for the qutrit depolarizing channel [15] are $E_{0,0} = \sqrt{1-\frac{8}{9}\lambda}\,I_3$ and $E_{s,t} = \frac{\sqrt{\lambda}}{3}X^sZ^t$. The state ρ in operator-sum representation is $\rho_2 = \sum_{r,s\in\mathbb{F}_3}E_{s,t}\rho E_{s,t}^\dagger$. On comparing ρ_1 and ρ_2, we get $\lambda = \frac{9}{8}p$.



(a)

(b)

(c)

FIG. 3. (a), (b) Show output noise probabilities $\epsilon^{(1)}_{\text{out}} = f(\epsilon_1, \epsilon_2)$ and $\epsilon^{(2)}_{\text{out}} = f(\epsilon_1, \epsilon_2)$ in green surfaces as a function of the input noise probabilities ϵ_1 and ϵ_2 when the MSD protocol is executed for distilling $|H_\pm\rangle$ state using the five-qutrit code [10]. The blue plots $\epsilon^{(1)}_{\text{out}} = \epsilon_1$ and $\epsilon^{(2)}_{\text{out}} = \epsilon_2$ are linear in ϵ_1 and ϵ_2 and intersect with the green plot at regions of threshold beyond which distillation is not possible. (c) Indicates that the threshold is around 0.21.

distillable, its quantum entropy increases and saturates to that of the maximally mixed case over iterations. The states $|H_{\pm 1}\rangle$ are distillable using five-qutrit code, so their entropy decreases to zero. The entropic observations are shown in Fig. 4. It is interesting to note that the probability of projecting the noisy quantum state over the code space projection operator is monotonically nondecreasing with the number of iterations during the execution of the MSD procedure when the state is distillable.

This can be proved by applying the MSD protocol iteratively over an n-length quantum code, satisfying the distillable conditions described in Sec. II B. First, we consider the input state to be distillable and prove that the probability of projection is increasing over iterations. Next, we consider the probability of projection to be increasing over iterations and prove that the state becomes distillable, implying the proof in both ways. We prove this property in Theorem 2.

Theorem 2. If the MSD procedure operates below the noise threshold to perform purification, the probability of projection onto the code space is a monotone nondecreasing function over iterations.

We prove Theorem 2 in Appendix A.

A. Scaling of the noise threshold based on the minimum distance of the quantum code

Consider a quantum stabilizer code $[[n, 1, d]]_p$ for distilling a state $|u\rangle$, which is an eigenstate of an operator U. The transversal logical Clifford operators satisfying triorthogonal conditions [16,17] are of the form $\bar{U} = U^{\otimes n}$. The stabilizer equivalent [1] of the operator $\bar{U}$ is $\tilde{U} = U_A \otimes I_B$.[5] Sets A and B contain the qudit positions, where operators U and I are applied on the qudits in a length-n code such that $A \cup B = \{1, 2, \dots, n\}$. The cardinality of the sets A and B are d and $n - d$, respectively, since the distance of the quantum code is the minimum symplectic weight among all logical operators

[5] The example can be the seven-qubit Steane code that has transversal logical gates as $X^{\otimes 7}$ and $Z^{\otimes 7}$. The stabilizer equivalent of these gates are $\bar{X} = I^{\otimes 4} \otimes X^{\otimes 3}$ and $\bar{Z} = I^{\otimes 4} \otimes Z^{\otimes 3}$ with symplectic operator weight 3 equal to code distance. It is easy to infer that the Clifford or non-Clifford gate should be of the form $I^{\otimes 4} \otimes U^{\otimes 3}$, and they can be represented as $U_{\{5,6,7\}} \otimes I_{\{1,2,3,4\}}$. For example, the logical Hadamard gate for the seven-qubit Steane code is given by $\bar{H} = I^{\otimes 4} \otimes H^{\otimes 3}$ as $\bar{H}\bar{X}\bar{H}^\dagger = \bar{Z}$ and $\bar{H}\bar{Z}\bar{H}^\dagger = \bar{X}$.

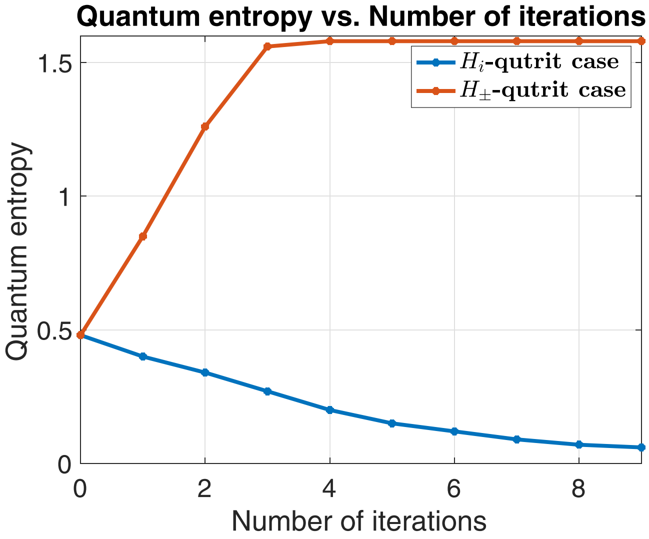


FIG. 4. Quantum entropy of the output magic states (eigenstates of the qutrit Hadamard gate), distilled through the code Q_1, is plotted against the iterations of the MSD protocol. Since the state $|H_i\rangle$ is not distillable using Q_1, therefore, the quantum entropy of the state $|H_i\rangle$ is increasing and saturated at the quantum entropy of the maximally mixed state $\log_2(3) \approx 1.598$ as shown in the red line. The blue line shows the quantum entropy of the $|H_{\pm 1}\rangle$ states that are distillable by the five-qutrit code. Therefore, entropies of $|H_\pm\rangle$ are tending towards zero with repeated application of the MSD procedure.

of the code [1]. Using an appropriate twirling scheme

$$\frac{1}{m}\left[\sum_{i=0}^{m-1} U^i \rho U^{i\dagger}\right],$$

such that $U^m = I$, the parameters of the state ρ can be reduced to ϵ such that

$$\rho = (1-\epsilon)|u_0\rangle\langle u_0| + \frac{\epsilon}{p-1}\sum_{i\in S_p\setminus\{0\}} |u_i\rangle\langle u_i|,$$

where $|u_i\rangle$, $\forall\, i \in S_p = \{0, 1, \dots, p-1\}$, are the eigenvectors of U with eigenvalues c_i.

To execute MSD an initial state ρ_n is prepared consisting of n qudits such that the qudit positions $q_i \in A$ over state ρ are denoted as ρ_A and leftover qudits whose positions $q_j \in B$ are in the state $|0\rangle$ denoted as $|0\rangle\langle 0|_B$. We represent ρ_n as

$$\rho_n = \rho_A \otimes |0\rangle\langle 0|_B, \tag{27}$$

where $\rho_A = \sum_{\underline{y}\in\mathbb{F}_p^{|A|}} (1-\epsilon)^{|A|-|\underline{y}|_H} (\frac{\epsilon}{p-1})^{|\underline{y}|_H} |u_{\underline{y}}\rangle\langle u_{\underline{y}}|$ and $|0\rangle\langle 0|_B = |0\rangle\langle 0|^{\otimes|B|}$ such that $|u_{\underline{y}}\rangle = |u_{y_1}\rangle \otimes |u_{y_2}\rangle \otimes \cdots \otimes |u_{y_{|A|}}\rangle$, for all $\underline{y} = (y_1, y_2, \dots, y_{|A|}) \in \mathbb{F}_p^{|A|}$, and $|\underline{y}|_H$ denotes the Hamming weight of $\underline{y}$. Equation (27) can be written as

$$\rho_n = \sum_{\underline{y}\in\mathbb{F}_p^{|A|}} (1-\epsilon)^{|A|-|\underline{y}|_H}\left(\frac{\epsilon}{p-1}\right)^{|\underline{y}|_H} \times |u_{\underline{y}}\rangle|0\rangle^{\otimes|B|}\langle u_{\underline{y}}|\langle 0|^{\otimes|B|}. \tag{28}$$

The projection operator of the code is given by [10]

$$\mathrm{P} = \frac{1}{p^{n-1}}\prod_{i=1}^{n-1}\sum_{j=0}^{p-1} g_i^j,$$

where $g_1, g_2, \dots, g_{n-1}$ are stabilizer generators of the $[[n, 1, d]]_p$ code. The next step is to project the state in Eq. (28) onto the code space as follows:

$$\rho_c = \frac{1}{P_s}\sum_{\underline{y}\in\mathbb{F}_p^{|A|}} (1-\epsilon)^{|A|-|\underline{y}|_H}\left(\frac{\epsilon}{p-1}\right)^{|\underline{y}|_H} \mathrm{P}|w_{\underline{y}}\rangle\langle w_{\underline{y}}|\mathrm{P}\,, \tag{29}$$

where $P_s = \mathrm{tr}(\mathrm{P}\rho_n)$ is the projection probability of the state ρ_n and $|w_{\underline{y}}\rangle = |u_{\underline{y}}\rangle|0\rangle^{\otimes|B|}$. Let C_i be the set containing tuples $\underline{y} \in \mathbb{F}_p^{|A|}$ so that $\mathrm{P}|w_{\underline{y}}\rangle$ is proportional to eigenvector $|\tilde{u}_i\rangle$ of $\tilde{U}$, $\forall\, i \in S_p$. Then, Eq. (29) can be written as

$$\rho_c = \sum_{\underline{x}\in C_0}(1-\epsilon)^{|A|-|\underline{x}|_H}\left(\frac{\epsilon}{p-1}\right)^{|\underline{x}|_H}\frac{a_{\underline{x}}^2}{P_s}|\tilde{u}_0\rangle\langle\tilde{u}_0| + \sum_{i\in S_p\setminus\{0\},\underline{x}\in C_i}(1-\epsilon)^{|A|-|\underline{x}|_H}\left(\frac{\epsilon}{p-1}\right)^{|\underline{x}|_H}\frac{b_{i,\underline{x}}^2}{P_s}|\tilde{u}_i\rangle\langle\tilde{u}_i|, \tag{30}$$

where $a_{\underline{x}}^2$, and $b_{i,\underline{x}}^2$ are probabilities of projection of $\mathrm{P}|w_{\underline{x}}\rangle$ for $\underline{x} \in C_0$ and $\mathrm{P}|w_{\underline{x}}\rangle$ for $\underline{x} \in C_i$.

Finally, the codeword state ρ_c is decoded using a decoding operator D_s such that $D_s|\tilde{u}_i\rangle\langle\tilde{u}_i|D_s^\dagger = |u_i\rangle\langle u_i| \otimes |0\rangle\langle 0|^{\otimes(N-1)}$ as follows:

$$D_s\rho_c D_s^\dagger = \sum_{\underline{x}\in C_0}(1-\epsilon)^{|A|-|\underline{x}|_H}\left(\frac{\epsilon}{p-1}\right)^{|\underline{x}|_H}\frac{a_{\underline{x}}^2}{P_s} \times |u_0\rangle\langle u_0| \otimes |0\rangle\langle 0|^{\otimes(n-1)} + \sum_{i\in S_p\setminus\{0\},\underline{x}\in C_i}(1-\epsilon)^{|A|-|\underline{x}|_H}\left(\frac{\epsilon}{p-1}\right)^{|\underline{x}|_H}\frac{b_{i,\underline{x}}^2}{P_s} \times |u_i\rangle\langle u_i| \otimes |0\rangle\langle 0|^{\otimes(n-1)}. \tag{31}$$

On discarding the last $n-1$ ancillae in the $|0\rangle$ state, we get a single logical qudit state

$$\rho_{\text{out}} = \sum_{\underline{x}\in C_0}(1-\epsilon)^{|A|-|\underline{x}|_H}\left(\frac{\epsilon}{p-1}\right)^{|\underline{x}|_H}\frac{a_{\underline{x}}^2}{P_s}|u_0\rangle\langle u_0| + \sum_{i\in S_p\setminus\{0\},\underline{x}\in C_i}(1-\epsilon)^{|A|-|\underline{x}|_H}\left(\frac{\epsilon}{p-1}\right)^{|\underline{x}|_H}\frac{b_{i,\underline{x}}^2}{P_s}|u_i\rangle\langle u_i|. \tag{32}$$

In general, ρ_{out} can be written as

$$\rho_{\text{out}} = (1-\epsilon_{\text{out}})|u_0\rangle\langle u_0| + \sum_{i\in S_p\setminus\{0\}}\epsilon_i|u_i\rangle\langle u_i|, \tag{33}$$

where $\epsilon_{\text{out}} = \sum_{i\in S_p\backslash\{0\}} \epsilon_i$. Comparing Eqs. (32) and (33), we get

$$\epsilon_{\text{out}} = \frac{1}{P_s}\sum_{i\in S_p\backslash\{0\}} \left(\frac{\epsilon}{p-1}\right)^{|A|} p_i + \sum_{i\in S_p\backslash\{0\}, y\in X_i} (1-\epsilon)^{|A|-y}\left(\frac{\epsilon}{p-1}\right)^{y}\frac{\bar{b}_{i,y}}{P_s}, \tag{34}$$

where $\bar{b}_{i,y} = \sum_{\underline{x}:|\underline{x}|_H=y} b_{i,\underline{x}}^2$, $p_i = \bar{b}_{i,|A|}$, $X_i = \{|\underline{x}|_H \mid \underline{x}\in C_i\backslash W_i\}$, and $W_i = \{\underline{x}|\, \underline{x}\in C_i, |\underline{x}|_H = |A|\}$. Using the projection equation provided in Sec. II B, we get

$$\sum_{i\in S_p} |\tilde{u}_i\rangle\langle\tilde{u}_i| = \sum_{\underline{y}\in\mathbb{F}_p^{|A|}} \text{P}|u_{\underline{y}}\rangle|0\rangle^{\otimes|B|}\langle u_{\underline{y}}|\langle 0|^{\otimes|B|}\text{P}. \tag{35}$$

Since $\text{P}|u_{\underline{x}}\rangle|0\rangle^{\otimes|B|} = a_{\underline{x}}|\tilde{u}_0\rangle$ for all $\underline{x}\in C_0$, and $\text{P}|u_{\underline{x}}\rangle|0\rangle^{\otimes|B|} = b_{i,\underline{x}}|\tilde{u}_i\rangle$ for all $\underline{x}\in C_i$, Eq. (35) can be simplified as

$$\sum_{i\in S_p} |\tilde{u}_i\rangle\langle\tilde{u}_i| = \sum_{\underline{x}\in C_0} a_{\underline{x}}^2|\tilde{u}_0\rangle\langle\tilde{u}_0| + \sum_{i\in S_p\backslash\{0\}, \underline{x}\in C_i} b_{i,\underline{x}}^2|\tilde{u}_i\rangle\langle\tilde{u}_i|. \tag{36}$$

Equating the terms of Eq. (36), we have

$$|\tilde{u}_i\rangle\langle\tilde{u}_i| = \sum_{\underline{x}\in C_i} b_{i,\underline{x}}^2|\tilde{u}_i\rangle\langle\tilde{u}_i|,\ \forall\, i\in S_p\backslash\{0\}. \tag{37}$$

On comparing coefficients of $|\tilde{u}_i\rangle\langle\tilde{u}_i|$ in Eq. (37), we get

$$\sum_{y\in X_i} \bar{b}_{i,y} = 1 - p_i,\ \forall\, i\in S_p. \tag{38}$$

Using Eq. (38) and the fact $|A| = d$, we can simplify Eq. (34) as

$$\epsilon_{\text{out}} \leqslant \frac{1}{P_s}\sum_{i\in S_p\backslash\{0\}} \left(\frac{\epsilon}{p-1}\right)^{d} p_i + \frac{1}{P_s}\sum_{i\in S_p\backslash\{0\}} (1-p_i)(1-\epsilon)^d f(i,\epsilon), \tag{39}$$

where $f(i,\epsilon) = \sum_{y\in X_i} (\frac{\epsilon}{(1-\epsilon)(p-1)})^y$. From Eq. (39), it is clear that ϵ_{out} decreases with increasing in distance d as $(\frac{\epsilon}{p-1})$ and $1-\epsilon$ both are less than 1.

Corollary 1. The length of a $[[n,1,d]]_p$ quantum code scales with the noise thresholds ϵ_{th} of the MSD protocol as follows:

$$n \geqslant 1 + 2\left[\ln\left(\frac{\epsilon_{\text{th}}P_s}{(p-1-\tilde{p})\tilde{f}(\epsilon_{\text{th}})}\right)\frac{1}{\ln(1-\epsilon_{\text{th}})} - 1\right], \tag{40}$$

where $\tilde{p} = \sum_{i\in S_p\backslash\{0\}} p_i$ and $\tilde{f}(\epsilon_{\text{th}}) = \max_i f(i,\epsilon_{\text{th}})$.

Proof. For large p and d, the term $(\frac{\epsilon}{p-1})^d \rightarrow 0$. Therefore, Eq. (39) can be approximated for $\epsilon_{\text{out}} = \epsilon = \epsilon_{\text{th}}$ as follows:

$$\sum_{i\in S_p\backslash\{0\}} (1-p_i)(1-\epsilon_{\text{th}})^d f(i,\epsilon_{\text{th}}) \geqslant \epsilon_{\text{th}}P_s. \tag{41}$$

Further simplifying Eq. (41), we get

$$(1-\epsilon_{\text{th}})^d \geqslant \frac{\epsilon_{\text{th}}P_s}{\sum_{i\in S_p\backslash\{0\}}(1-p_i)f(i,\epsilon_{\text{th}})}. \tag{42}$$

Finally, the distance d can be lower bounded as follows:

$$d \geqslant \frac{1}{\ln(1-\epsilon_{\text{th}})}\ln\left(\frac{\epsilon_{\text{th}}P_s}{\sum_{i\in S_p\backslash\{0\}}(1-p_i)f(i,\epsilon_{\text{th}})}\right). \tag{43}$$

Using the quantum Singleton bound [18], we can lower bound the code length of the $[[n,1,d]]_p$ quantum code as

$$n \geqslant 1 + 2\left[\ln\left(\frac{\epsilon_{\text{th}}P_s}{\sum_{i\in S_p\backslash\{0\}}(1-p_i)f(i,\epsilon_{\text{th}})}\right) \times\frac{1}{\ln(1-\epsilon_{\text{th}})} - 1\right]. \tag{44}$$

Let $\tilde{f}(\epsilon_{\text{th}}) = \max_i f(i,\epsilon_{\text{th}})$, then Eq. (44) becomes

$$n \geqslant 1 + 2\left[\ln\left(\frac{\epsilon_{\text{th}}P_s}{(p-1-\tilde{p})\tilde{f}(\epsilon_{\text{th}})}\right)\frac{1}{\ln(1-\epsilon_{\text{th}})} - 1\right],$$

where $\tilde{p} = \sum_{i\in S_p\backslash\{0\}} p_i$. ■

From Corollary 1, we can easily calculate the scaling of the code length from n_1 to n_2 with a 1% increase in the noise threshold from $\epsilon_{\text{th}}^{(1)}$ to $\epsilon_{\text{th}}^{(2)}$ such that $\epsilon_{\text{th}}^{(2)} = 1.01\epsilon_{\text{th}}^{(1)}$ while considering parameters P_S, p_i, and p constant. Using Eq. (40), we get

$$n_2 - n_1 \geqslant 2\left[\ln\left(\frac{1.01\epsilon_{\text{th}}^{(1)}P_s}{\sum_{i\in S_p\backslash\{0\}}(1-p_i)f\left(i,1.01\epsilon_{\text{th}}^{(1)}\right)}\right) \times\frac{1}{\ln\left(1-1.01\epsilon_{\text{th}}^{(1)}\right)} - \frac{1}{\ln\left(1-\epsilon_{\text{th}}^{(1)}\right)} \times\ln\left(\frac{\epsilon_{\text{th}}^{(1)}P_s}{\sum_{i\in S_p\backslash\{0\}}(1-p_i)f\left(i,\epsilon_{\text{th}}^{(1)}\right)}\right)\right]. \tag{45}$$

Using the fact $\frac{1}{\ln(1-1.01\epsilon_{\text{th}}^{(1)})} > \frac{1}{\ln(1-\epsilon_{\text{th}}^{(1)})}$, we get

$$n_2 - n_1 > \frac{2}{\ln\left(1-\epsilon_{\text{th}}^{(1)}\right)}\ln\left(\frac{1.01\sum_{i\in S_p\backslash\{0\}}(1-p_i)f\left(i,\epsilon_{\text{th}}^{(1)}\right)}{\sum_{i\in S_p\backslash\{0\}}(1-p_i)f\left(i,1.01\epsilon_{\text{th}}^{(1)}\right)}\right). \tag{46}$$

Similarly, using Eq. (43), we can calculate the scaling of the code distance from d_1 to d_2 for 1% scaling of the noise threshold as follows:

$$d_2 - d_1 > \frac{1}{\ln\left(1-\epsilon_{\text{th}}^{(1)}\right)}\ln\left(\frac{1.01\sum_{i\in S_p\backslash\{0\}}(1-p_i)f\left(i,\epsilon_{\text{th}}^{(1)}\right)}{\sum_{i\in S_p\backslash\{0\}}(1-p_i)f\left(i,1.01\epsilon_{\text{th}}^{(1)}\right)}\right). \tag{47}$$

IV. PRACTICAL DISTILLABLE CODES

In this section, we provide some practical distillable codes to distill $|T\rangle$ and $|S\rangle$ states with improved noise thresholds as compared to the codes using 5-qubit code and 11-qutrit Golay code [6,7]. As $|T\rangle$ and $|S\rangle$ states defined in Sec. I are most distant from the Wigner polytope [6,7,10,19], both the states can be distilled with higher thresholds compared to other states.

A. Distillation of $|T\rangle$ state using 31-qubit quantum BCH code

Consider $C = [n, k, d]$ classical Bose-Chaudhuri-Hocquenghem (BCH) code over $\mathbb{F}_p^m$ with generator polynomial [20,21]

$$g(x) = \prod_{w \in \mathrm{W}_C} (x - \alpha^w),$$

where α is the primitive element and W_C is a union of cyclotomic cosets $C_\beta = \{\beta p^i \bmod (n)\}$ for $i \in \{0, 1, \ldots, m-1\}$ and $\beta \in \mathbb{F}_p^m$. Similarly, the generator polynomial of the dual code of C denoted as $C^\perp$ can be given as

$$h(x) = \prod_{v \in \mathrm{W}_{C^\perp}} (x - \alpha^v),$$

where $\mathrm{W}_{C^\perp} = \{-w \bmod (n) | w \in \{0, 1, \ldots, n-1\} \backslash \mathrm{W}_C\}$. To design a quantum BCH code using CSS construction [1], the code C should satisfy the weakly self-dual containing property, i.e., $\mathrm{W}_{C^\perp} \subseteq \mathrm{W}_C$ [20]. For example, a $[31, 16]_2$ classical BCH code over $\mathbb{F}_{32}$ with the design distance of 7 is a self-dual code as the cyclotomic cosets $\mathrm{W}_C = \mathrm{W}_{C^\perp} = C_1 \cup C_3 \cup C_5$. The parity check matrix of the code is given by [20,22]

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \ldots \alpha^{30} \\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18} \ldots \alpha^{28} \\ 1 & \alpha^5 & \alpha^{10} & \alpha^{15} & \alpha^{20} & \alpha^{25} & \alpha^{30} \ldots \alpha^{26} \end{bmatrix}, \tag{48}$$

where α is the primitive element with primitive polynomial $x^5 + x^2 + 1$. Using the CSS construction, a $[[31, 1, 16]]_2$ quantum BCH code can be created with stabilizer matrix given by

$$H_{\mathrm{CSS}} = \begin{bmatrix} H & 0_{15\times 31} \\ 0_{15\times 31} & H \end{bmatrix},$$

where $0_{15\times 31}$ is a zero matrix of dimension 15×31. By construction, the matrix H is full rank, leading to a distance of 16. The logical-X operator is denoted as $\bar{X} = X^{\otimes 31}$, and logical-Z operator is $\bar{Z} = Z^{\otimes 31}$. The operators $\bar{X}$ and $\bar{Z}$ are transversal since the isomorphic vector $(1, 1, \ldots, 1)$ to the operators $\bar{X}$ and $\bar{Z}$ does not belong to the row space but to the null space of the matrix H in $\mathbb{F}_2$ [1]. As $\bar{X}$ and $\bar{Z}$ gates are both transversal, logical-T operator [6] is also transversal denoted as $\bar{T} = T^{\otimes 31}$ since $\bar{T}\bar{X}\bar{T}^\dagger = \bar{Y}$, $\bar{T}\bar{Y}\bar{T}^\dagger = \bar{Z}$, and $\bar{T}\bar{Z}\bar{T}^\dagger = \bar{X}$, where $T = \frac{e^{i\pi/4}}{\sqrt{2}}[\begin{smallmatrix} 1 & 1 \\ i & -i \end{smallmatrix}]$. The $|T\rangle$ state is an eigenstate of the T gate [6], i.e., $T|T\rangle = e^{i\pi/3}|T\rangle$ and so $|\tilde{T}\rangle$ is also an eigenstate of T such that $T|\tilde{T}\rangle = e^{-i\pi/3}|\tilde{T}\rangle$.

The basic requirement of MSD is satisfied by the 31-qubit BCH code, which means $p_0 \neq 0$ as described in Sec. II B. As the logical-$\bar{T}$ operator and the projector operator P commute since $[\bar{T}, S_i] = 0$, $\forall\, i \in \{0, 1, \ldots, 30\}$, where S_i are the stabilizer generators isomorphic to the rows of the H_{CSS} matrix, we have

$$\bar{T}P|T\rangle^{\otimes 31} = e^{i\pi/3}P|T\rangle^{\otimes 31}, \tag{49}$$

where

$$P = \frac{1}{2^{30}}\Pi_{i=1}^{30}(I + S_i). \tag{50}$$

To distill the state $|T\rangle$, prepare a state ρ and then apply a dephasing transformation provided in [6] to reduce the state parameters, so that the state ρ becomes

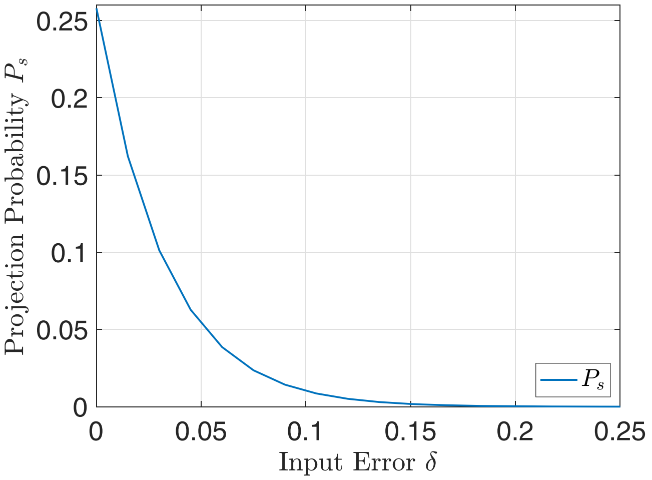


FIG. 5. The projection probability P_s increases as the noise probability δ decreases over iterations of the MSD protocol for the $[[31, 1, 16]]_2$ quantum BCH code.

$$\rho = (1-\delta)|T\rangle\langle T| + \delta|\tilde{T}\rangle\langle\tilde{T}|. \tag{51}$$

Now, the MSD procedure described in Sec. II can be applied by preparing 31 copies of the state ρ and projecting it onto the code space using a projection operator. The prepared 31-qubit input state $\rho^{\otimes 31}$ is given by

$$\rho_{\mathrm{in}} = \sum_{\underline{x} \in \mathbb{F}_2^{31}} (1-\delta)^{(31-|\underline{x}|_H)}\delta^{|\underline{x}|_H}|T_{\underline{x}}\rangle\langle T_{\underline{x}}|, \tag{52}$$

where $|T_{\underline{x}}\rangle = |T_{x_1}\rangle \otimes |T_{x_2}\rangle \otimes \cdots \otimes |T_{x_{31}}\rangle$ for all $\underline{x} = (x_1, x_2, \ldots, x_{31})$ such that $|T_0\rangle = |T\rangle$ and $|T_1\rangle = |\tilde{T}\rangle$, and $|\underline{x}|_H$ is the Hamming weight of $\underline{x}$. On projecting the state ρ_{in} onto the code space using the projection operator in Eq. (50), we get

$$\rho_c = \frac{1}{P_s}\sum_{\underline{x} \in \mathbb{F}_2^{31}} (1-\delta)^{(31-|\underline{x}|_H)}\delta^{|\underline{x}|_H}P|T_{\underline{x}}\rangle\langle T_{\underline{x}}|P, \tag{53}$$

where $P_s = \mathrm{tr}(\rho_{\mathrm{in}}P)$. P_s improves as the noise probability δ decreases over iterations as shown in Fig. 5, proved in Theorem 2. Now, if $|L_{\underline{x}}\rangle = P|T_{\underline{x}}\rangle$ belongs to the code space, then $|L_{\underline{x}}\rangle$ should be an eigenvector of logical-T operator, i.e.,

$$\bar{T}|L_{\underline{x}}\rangle = e^{i\frac{\pi}{3}(31-2|\underline{x}|_H)}|L_{\underline{x}}\rangle. \tag{54}$$

From Eq. (54), it is clear that for $|\underline{x}|_H \in A = \{2, 5, 8, \ldots, 29\}$, $|L_{\underline{x}}\rangle$ is not an eigenvector of logical-T gate; therefore, $|L_{\underline{x}}\rangle = 0$. Similarly, if $|\underline{x}|_H \in B = \{0, 3, 6, \ldots, 30\}$, then $|L_{\underline{x}}\rangle$ is proportional to the $|\bar{T}_0\rangle$ eigenvector of logical-T gate with eigenvalue $e^{i\pi/3}$.

Finally, if $|\underline{x}|_H \in C = S_{31}\backslash(A \cup B)$, where $S_{31} = \{0, 1, 2, \ldots, 31\}$, then $|L_{\underline{x}}\rangle$ is proportional to the $|\bar{T}_1\rangle$ eigenvector of logical-T gate with eigenvalue $e^{-i\pi/3}$. Thus, Eq. (53) can be simplified as

$$\rho_c = \frac{1}{P_s}\sum_{y \in B}(1-\delta)^{(31-y)}\delta^y a_y|\bar{T}_0\rangle\langle\bar{T}_0| + \frac{1}{P_s}\sum_{y \in C}(1-\delta)^{(31-y)}\delta^y b_y|\bar{T}_1\rangle\langle\bar{T}_1|, \tag{55}$$

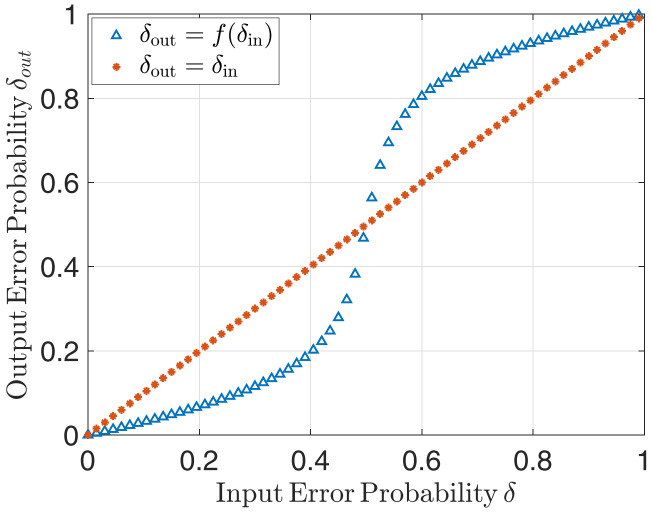


FIG. 6. The relation between δ_{out} and δ_{in} is shown in the blue line $\delta_{\mathrm{out}} = f(\delta_{\mathrm{in}})$, and the reference line $\delta_{\mathrm{out}} = \delta_{\mathrm{in}}$ is shown in the red line. Both the lines intersect at a noise threshold of 0.498, below which the noisy $|T\rangle$ state can be distilled using the $[[31, 1, 16]]_2$ code.

where $a_{|\underline{x}|_H}$ and $b_{|\underline{x}|_H}$ are the constants of proportionality such that $\sum_{\underline{x}:|\underline{x}|_H \in B} |L_{\underline{x}}\rangle = \sqrt{a_{|\underline{x}|_H}}|\bar{T}_0\rangle$ and $\sum_{\underline{x}:|\underline{x}|_H \in C} |L_{\underline{x}}\rangle = \sqrt{b_{|\underline{x}|_H}}|\bar{T}_1\rangle$.

Finally, the state ρ_c can be decoded to obtain

$$\rho_{\mathrm{out}} = (1-\delta_{\mathrm{out}})|T\rangle\langle T| + \delta_{\mathrm{out}}|\tilde{T}\rangle\langle\tilde{T}|. \tag{56}$$

The relation between δ_{out} vs δ is shown in Fig. 6, with a noise threshold of around 0.498, which is greater than the noise threshold of 0.173, using the $[[5, 1, 3]]_2$ code [6].

B. 13-qutrit code distillation of $|S\rangle$ state

In this section, we discuss the distillation of state $|S\rangle$, which is an eigenstate of the qutrit Hadamard gate

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{bmatrix},$$

where $\omega = e^{i\frac{2\pi}{3}}$, with eigenvalue of i [7] using a $[13, 7, 7]_3$ classical ternary code C_3 with parity check matrix

$$H_w = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 2 & 1 & 2 & 1 & 2 & 1 & 2 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 2 & 2 \\ 0 & 2 & 2 & 0 & 0 & 0 & 2 & 2 & 1 & 1 & 2 & 2 & 1 \\ 0 & 0 & 2 & 2 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 0 & 2 & 0 & 2 & 0 & 0 & 2 & 1 & 2 & 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 2 & 2 & 2 & 2 & 1 & 1 \end{bmatrix}.$$

As code C_3 is self-dual since $H_w H_w^{\mathrm{T}} = 0 \bmod (3)$ so a $[[13, 1, 7]]_3$ quantum CSS code can be constructed using the classical code C_3 with stabilizer generator matrix given by

$$H_{\mathrm{CSS}} = \begin{bmatrix} H_w & 0_{6\times 13} \\ 0_{6\times 13} & H_w \end{bmatrix},$$

where $0_{6\times 13}$ is a zero matrix of dimension 6×13. The transversal logical-X and logical-Z operators of the code are given by

$$\bar{X} = X^2 \otimes X^{\otimes 3} \otimes X^2 \otimes X^2 \otimes X^{\otimes 7},$$
$$\bar{Z} = Z^2 \otimes Z^{\otimes 3} \otimes Z^2 \otimes Z^2 \otimes Z^{\otimes 7},$$

since the vector $(2, 1, 1, 1, 2, 2, 1, \ldots, 1)$ isomorphic to $\bar{X}$ and $\bar{Z}$ is in the null space but not in the row space of H_w in $\mathbb{F}_3$. The logical-H operator is $\bar{H} = H^{\otimes 13}$ since $\bar{H}\bar{X}\bar{H} = \bar{Z}$ and $\bar{H}\bar{Z}\bar{H} = \bar{X}$. The eigenstates of H gate are $|S\rangle$, $|H_1\rangle$, and $|H_{-1}\rangle$ with eigenvalues i, 1, and -1. First, a state ρ is prepared and twirling scheme is applied given in [7]

$$\rho_1 = \frac{1}{4}\sum_{i=0}^{3} H^i \rho H^{-i}. \tag{57}$$

Again, on the state ρ_1, a new twirling scheme is applied such that the state is restricted to a single parameter α, i.e., [7]

$$\rho_2 = \tfrac{1}{2}[\rho_1 + \tilde{H}\rho_1\tilde{H}^{\dagger}], \tag{58}$$

where $\tilde{H} = \bar{i}|S\rangle\langle S| + e^{i3\pi/4}|H_{-1}\rangle\langle H_1| - e^{i\pi/4}|H_1\rangle\langle H_{-1}|$.

Finally, the state becomes

$$\rho_2 = (1-\alpha)|S\rangle\langle S| + \frac{\alpha}{2}(|H_1\rangle\langle H_1| + |H_{-1}\rangle\langle H_{-1}|). \tag{59}$$

Now, prepare 13 copies of the state ρ_2 as follows:

$$\rho_{13} = \sum_{\underline{x}\in\mathbb{F}_3^{13}} (1-\alpha)^{13-|\underline{x}|_H}\left(\frac{\alpha}{2}\right)^{|\underline{x}|_H} |H_{\underline{x}}\rangle\langle H_{\underline{x}}|. \tag{60}$$

After that project the state ρ_{13} onto the code space using a projection operator [10]

$$\Pi = \frac{1}{3^{12}}\sum_{\underline{x}\in\mathbb{F}_3^{12}} g_1^{x_1} g_2^{x_2} \ldots g_{12}^{x_{12}},$$

where stabilizer generators $g_1, g_2, \ldots, g_{12}$ are isomorphic to the rows of H_{CSS} matrix [1]. The necessary condition of distillation of the state $|S\rangle$ using the 13-qutrit code is satisfied as

$$\bar{H}\Pi|S\rangle^{\otimes 13} = i^{13}\Pi|S\rangle^{\otimes 13} = i\Pi|S\rangle^{\otimes 13}. \tag{61}$$

The projected state is given by

$$\tilde{\rho_{13}} = \frac{1}{P_s}\left[\sum_{\underline{x}\in\mathbb{F}_3^{13}} (1-\alpha)^{13-|\underline{x}|_H}\left(\frac{\alpha}{2}\right)^{|\underline{x}|_H} \Pi|H_{\underline{x}}\rangle\langle H_{\underline{x}}|\Pi\right], \tag{62}$$

where $|H_0\rangle = |S\rangle$, and $|H_{\underline{x}}\rangle = |H_{x_1}\rangle \otimes |H_{x_2}\rangle \otimes \cdots \otimes |H_{x_{13}}\rangle$. If $\Pi|H_{\underline{x}}\rangle$ belongs to the code space, it should be an eigenvector of $\bar{H}$ operator. We can verify that for all $\underline{x}$ having odd Hamming weight, $\Pi|H_{\underline{x}}\rangle$ is proportional to either $|\bar{H}_1\rangle$ or $|\bar{H}_{-1}\rangle$ eigenvectors of $\bar{H}$ operator with eigenvalues 1 and -1 since $\bar{H}\Pi|H_{\underline{x}}\rangle = \pm\Pi|H_{\underline{x}}\rangle$. For all $\underline{x}$ having even Hamming weight, $\Pi|H_{\underline{x}}\rangle$ is either proportional to $|\bar{S}\rangle$, which is eigenvector of $\bar{H}$ with eigenvalue i, or $\Pi|H_{\underline{x}}\rangle$ is out of the code space as $\bar{H}\Pi|H_{\underline{x}}\rangle = \pm i\Pi|H_{\underline{x}}\rangle$.

Therefore, Eq. (62) can be rewritten as

$$\tilde{\rho_{13}} = \frac{1}{P_s}\sum_{x\in S_e} a_x^2 (1-\alpha)^{13-x}\left(\frac{\alpha}{2}\right)^x |\bar{S}\rangle\langle\bar{S}| + \frac{1}{P_s}\sum_{x\in S_o} b_x^2 \times (1-\alpha)^{13-x}\left(\frac{\alpha}{2}\right)^x (|\bar{H}_1\rangle\langle\bar{H}_1| + |\bar{H}_{-1}\rangle\langle\bar{H}_{-1}|), \tag{63}$$

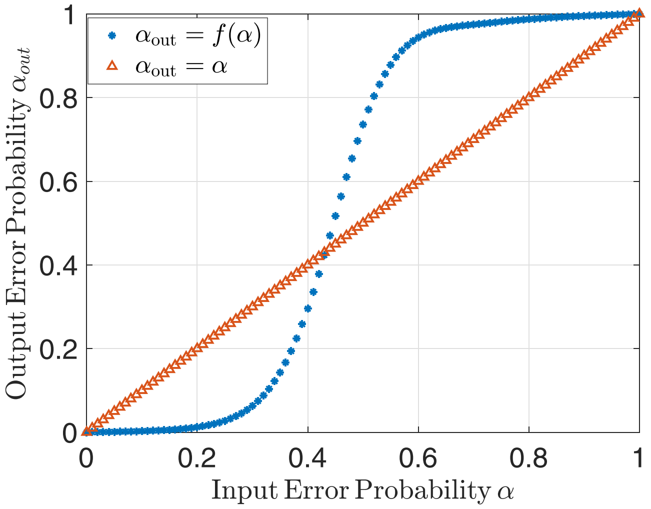


FIG. 7. The relation α_{out} vs α is shown with the blue line $\alpha_{\text{out}} = f(\alpha)$ for the 13-qutrit code. The red line is a reference line $\alpha_{\text{out}} = \alpha$ shown to intersect with the blue line around 0.425, a threshold point. The state is distillable if α is less than threshold point 0.425.

such that $S_e = \{|\underline{x}|_H \equiv 0 \bmod (2) : \underline{x} \in \mathbb{F}_3^{13}\}$, $S_o = \{|\underline{x}|_H \equiv 1 \bmod (2) : \underline{x} \in \mathbb{F}_3^{13}\}$, $a^2_{|\underline{x}|_H}$ and $b^2_{|\underline{x}|_H}$ are the sum of probabilities of projection of all the states $|H_{\underline{x}}\rangle$ with the particular Hamming weight onto the state $|\bar{S}\rangle$, $|\bar{H}_1\rangle$, and $|\bar{H}_{-1}\rangle$. Finally, we decode the codeword $\tilde{\rho_{13}}$ using a decoding operator W similar to what was done in Sec. III A, i.e., $W|\bar{S}\rangle\langle\bar{S}|W^\dagger = |S\rangle\langle S| \otimes |0\rangle\langle 0|^{\otimes 12}$ and $W|\bar{H}_{\pm 1}\rangle\langle\bar{H}_{\pm 1}|W^\dagger = |H_{\pm 1}\rangle\langle H_{\pm 1}| \otimes |0\rangle\langle 0|^{\otimes 12}$. We obtain $\tilde{\rho} = W\tilde{\rho_{13}}W^\dagger = \rho_{\text{out}} \otimes |0\rangle\langle 0|^{\otimes 12}$, where

$$\rho_{\text{out}} = (1-\alpha_{\text{out}})|S\rangle\langle S| + \left(\frac{\alpha_{\text{out}}}{2}\right)(|H_1\rangle\langle H_1| + |H_{-1}\rangle\langle H_{-1}|). \tag{64}$$

On simplification and discarding last 12 qutrits in state $\tilde{\rho}$, we get

$$\alpha_{\text{out}} = \frac{1}{P_s}\sum_{y\in S_o} b_y^2 (1-\alpha)^{13-y}\left(\frac{\alpha}{2}\right)^y. \tag{65}$$

From Fig. 7, we conclude that the state $|S\rangle$ is distillable using a [[13, 1, 7]]$_3$ qutrit code up to the noise threshold of 0.425.

C. Improvements on the code noise threshold of the $|S\rangle$ state: A case study

In this section, we discuss improving the noise threshold by increasing the distance of the quantum code. The upper bound on the noise threshold of the state $|S\rangle$, after which state $|S\rangle$ is not useful for MSD, can be calculated using the discrete Wigner function (DWF) [7]. For a r-qudit state ρ_r, the DWF, i.e., $W(\underline{x}, \underline{z})^{(r)}$, for all $\underline{x}, \underline{z} \in \mathbb{Z}_p^r$, is a quasiprobability distribution calculated using phase-point operators:

$$S^{(r)}_{\underline{0},\underline{0}} = \sum_{\underline{x},\underline{z}\in\mathbb{Z}_p^r} \omega^{\underline{x}\cdot\underline{z}2^{-1}} Z^{\underline{z}} X^{\underline{x}},$$

$$S^{(r)}_{\underline{x},\underline{z}} = Z^{\underline{z}} X^{\underline{x}} S^{(r)}_{\underline{0},\underline{0}} (Z^{\underline{z}} X^{\underline{x}})^\dagger,$$

where $Z^{\underline{z}} X^{\underline{x}} = Z^{z_1}X^{x_1} \otimes Z^{z_2}X^{x_2} \otimes \cdots Z^{z_r}X^{x_r}$, so $W(x,z)^{(r)}$ is given by

$$W(\underline{x}, \underline{z})^{(r)} = \frac{1}{p^2}\text{tr}\big(\rho_r S^{(r)}_{\underline{x},\underline{z}}\big). \tag{66}$$

For the single-qutrit ($r = 1$) noisy state ρ in Eq. (59), the DWF is calculated as follows:

$$W(x,z,\alpha)^{(1)} = \begin{cases} -\frac{1}{3} + \frac{4}{9}\alpha, & (x,z) = (0,0) \\ \frac{1}{6} - \frac{1}{18}\alpha, & (x,z) \neq (0,0). \end{cases} \tag{67}$$

The state ρ is not magic state distillable if $W(x,z,\alpha)^{(1)}$ is non-negative due to the necessity of contextuality required for MSD [7,23,24] as the classical simulability of the Clifford operations and stabilizer generators measurement is feasible beyond the upper bound of the magic state set by contextuality, so the state has no quantum advantage. $W(x,z,\alpha)^{(1)}$, $\forall\, (x,z) \in \mathbb{Z}_3 \times \mathbb{Z}_3$, is non-negative for $\alpha \geqslant 0.75$; therefore, the noise threshold of the state $|S\rangle$ is upper bounded by 0.75.

Below, we prove that the noise threshold over the depolarizing channel increases monotonically with the minimum distance. This can be proved by considering two different quantum distillable codes $Q_1 = [[n_1, 1, d_1]]_p$ and $Q_2 = [[n_2, 1, d_2]]_p$ such that $d_2 > d_1$ for some magic state $|u\rangle$. Using Eq. (39) for Q_1 and Q_2, we show that the noise threshold of Q_2, i.e., ϵ_2 is greater than the noise threshold of Q_1, i.e., ϵ_1 over the depolarizing channel.

Theorem 3. The depolarizing noise threshold of the magic state distilled using MSD increases monotonically with the distance of the quantum stabilizer code.

We prove Theorem 3 in Appendix A.

From Theorem 3, it is clear that the upper bound on the noise threshold of the code is achievable if the large-distance qutrit quantum codes are used to distill the state $|S\rangle$. An example of such a code is given below.

Example 2. In this example, we design a [[29, 1, 15]]$_3$ qutrit stabilizer code similar to the code described in Sec. IV B, but with improved code distance. Figure 8 shows the dual-containing code parity check matrix, which is distance 15 code. The distillation procedure of this code is similar to the procedure defined in Sec. IV B. Using the CSS formalism, a [[29, 1, 15]]$_3$ stabilizer code can be generated. The logical operators of the code are given as

$$\begin{aligned}\bar{X} =& X^2 \otimes X^{\otimes 2} \otimes X^2 \otimes X^{\otimes 2} \otimes X^2 \otimes X^{\otimes 6} \otimes X^2 \otimes X \\ &\otimes (X^2)^{\otimes 2} \otimes I \otimes X \otimes I \otimes X \otimes (X^2)^{\otimes 2} \otimes I \otimes (X^2)^{\otimes 3} \\ &\otimes X \otimes I,\end{aligned} \tag{68}$$

$$\begin{aligned}\bar{Z} =& Z^2 \otimes Z^{\otimes 2} \otimes Z^2 \otimes Z^{\otimes 2} \otimes Z^2 \otimes Z^{\otimes 6} \otimes Z^2 \otimes Z \\ &\otimes (Z^2)^{\otimes 2} \otimes I \otimes Z \otimes I \otimes Z \otimes (Z^2)^{\otimes 2} \otimes I \otimes (Z^2)^{\otimes 3} \\ &\otimes Z \otimes I\end{aligned} \tag{69}$$

for the same reason given for [[13, 1, 7]]$_3$ code such that $\bar{X}\bar{Z} = \omega\bar{Z}\bar{X}$, where $\omega = e^{i2\pi/3}$. Therefore, the logical Hadamard gate is $\bar{H} = H^{\otimes 29}$ as $\bar{H}\bar{X}\bar{H}^\dagger = \bar{Z}$ and $\bar{H}\bar{Z}\bar{H}^\dagger = \bar{X}$. The necessary condition for the distillation of $|S\rangle$ state can be checked as follows:

$$\bar{H}\Pi|S\rangle^{\otimes 29} = i^{29}\Pi|S\rangle^{\otimes 29} = i\Pi|S\rangle^{\otimes 29}.$$

$$\mathbf{H} = \begin{bmatrix}
1&1&1&0&0&0&0&0&0&0&0&0&0&0&2&1&2&1&2&1&2&1&2&1&2&1&2&1&2\\
0&0&1&1&1&0&0&0&0&0&0&0&0&0&1&2&2&1&1&2&2&1&1&2&2&1&1&2&2\\
0&1&0&0&1&1&0&0&0&0&0&0&0&0&2&2&1&1&2&2&1&1&2&2&1&1&2&2&1\\
1&0&0&1&0&1&0&0&0&0&0&0&0&0&1&1&1&2&2&2&2&1&1&1&1&2&2&2&2\\
0&1&0&1&0&0&1&0&0&0&0&0&0&0&2&1&2&2&1&2&1&1&2&1&2&2&1&2&1\\
0&0&1&0&0&1&1&0&0&1&1&1&0&0&1&2&2&2&2&1&1&1&1&2&2&2&2&1&1\\
0&0&1&0&0&1&1&0&1&0&0&1&1&0&2&2&1&2&1&1&2&1&2&2&1&2&1&1&2\\
0&0&1&0&0&1&1&1&0&0&1&0&1&0&1&1&1&1&1&1&1&2&2&2&2&2&2&2&2\\
0&0&1&0&0&1&1&0&1&0&1&0&0&1&2&1&2&1&2&1&2&2&1&2&1&2&1&2&1\\
0&0&1&0&0&1&1&1&0&0&0&1&0&1&1&2&2&1&1&2&2&2&1&1&2&2&1&1&2\\
1&0&0&0&1&0&1&0&0&0&0&0&0&0&2&2&1&1&2&2&1&2&2&1&1&2&2&1&1\\
0&0&1&0&0&1&1&1&1&1&0&0&0&0&1&1&1&2&2&2&2&2&2&2&2&1&1&1&1\\
0&0&1&0&0&1&1&0&0&1&0&0&1&1&2&1&2&2&1&2&1&2&1&2&1&1&2&1&2\\
1&0&2&0&0&0&0&2&2&0&2&0&1&1&1&0&1&0&1&2&1&2&0&1&1&0&0&0&0
\end{bmatrix}$$

FIG. 8. The parity check matrix of the $[29, 15]_3$ ternary self-dual code.

Hence, a similar process is repeated for the distillation of $|S\rangle$ state as mentioned in Sec. IV B.

The computation of the noise threshold for the $[[29, 1, 15]]_3$ code is computationally expensive, but using Eq. (39), we can calculate the lower bound on the noise threshold for such a code as $\epsilon_{\text{out}} = \epsilon$, $P_1 = 4.43e{-}9$, $P_2 = 2.44e{-}9$, and $P_s = 0.44e{-}13$. The distance of the code $[[29, 1, 15]]_3$ is 15. Therefore, the lower bound on the code threshold can be calculated as

$$F(\epsilon) = (\epsilon/2)^{15}(P_1 + P_2) + (2 - P_1 - P_2)(1-\epsilon)^{15} \times \sum_{i\in\{0,2,4,\dots,14\}} \left(\frac{\epsilon}{2(1-\epsilon)}\right)^i - \frac{\epsilon P_s}{2} \geqslant 0.$$

From Fig. 9, it is clear that the lower bound on the noise threshold ϵ^* is around 0.71, which is the value of ϵ for which $F(\epsilon)$ is minimum and positive. Clearly, the noise threshold of 0.71 obtained from $[[29, 1, 15]]_3$ is greater than the noise threshold of 0.387 achieved using the $[[11, 1, 5]]_3$ Golay code [7].

From Example 2, it is clear that the upper bound of 0.75 on the noise threshold of the state $|S\rangle$ is very close to achieving. However, it requires large length quantum codes. Using Eq. (40), we can calculate the lower bound on the code length for $f(i, 0.75) = \sum_{y\in X_i}(\frac{3}{2})^y = \frac{1.5^{(\lfloor \frac{D}{2}\rfloor+1)}-1}{1.25}$ as

$$n \geqslant 1 + 2\left[\ln\left(\frac{0.937P_s}{(2-p_1-p_2)(1.5^{(\lfloor \frac{D}{2}\rfloor+1)}-1)}\right)\frac{1}{\ln(0.25)} - 1\right]. \tag{70}$$

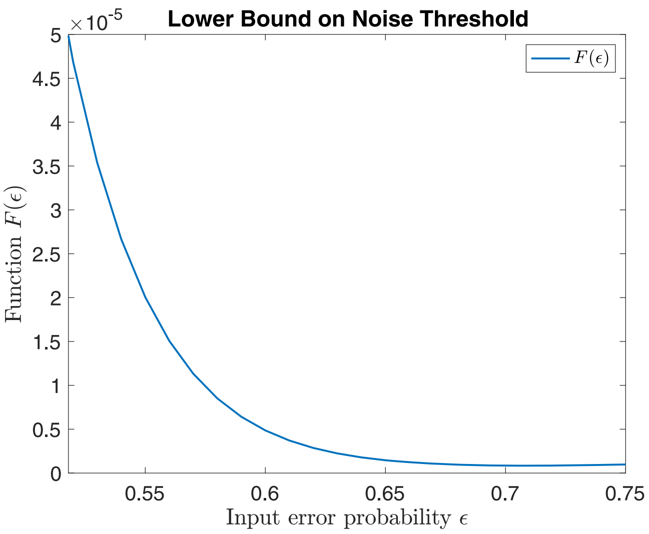


FIG. 9. The function $F(\epsilon)$ is plotted against the noise probability ϵ. The plot shows that $F(\epsilon)$ attains a minimum value when ϵ is around 0.71.

Further simplification will get us

$$n \geqslant 1 + 2\left[\ln\left(\frac{1.06(2-p_1-p_2)(1.5^{(\lfloor \frac{D}{2}\rfloor+1)}-1)}{P_s}\right) \times \frac{1}{\ln(4)} - 1\right], \tag{71}$$

where $X_i = \{0, 2, \dots, 2\lfloor \frac{D}{2}\rfloor\}$, $\forall\, i \in \{1, 2\}$.

In Table I, we summarize the results of this section and provide the comparisons of the noise thresholds with our proposed codes of the $|T\rangle$ in [6] and $|S\rangle$ in [7] states. Although the code parameters listed in Table I are specific examples, the reader must note that Corollary 1 shows how the code length and minimum distance scale to achieve noise thresholds arbitrary close to the computed theoretical limits.

So far we have provided $Q_1 = [[31, 1, 16]]_2$ qubit and $Q_2 = [[29, 1, 15]]_3$ qutrit codes for distilling the $|T\rangle$ and $|S\rangle$ states. We have so far not discussed the implementation of the decoders required in third step for MSD, described in Sec. II. In the next section, we provide the decoders (reverse of encoders) for any general $[[n, k, d]]_p$-qudit stabilizer code. The reader must note that although Q_1 and Q_2 encode only a single-qudit state, the construction of general decoders can be useful for new MSD techniques utilizing multiqudit codes to distill k magic states, $k = 1$ being a special case.

V. EFFICIENT ENCODING CIRCUITS OF A GENERAL QUANTUM CSS CODE FOR MSD

In this section, we provide algorithms to generate efficient encoding circuits for the quantum CSS and hypergraph product codes useful for the decoding step[6] of the MSD procedure. First, we briefly review the quantum CSS and hypergraph product codes and further provide a procedure to encode them.

A. Qudit quantum CSS codes

Consider two classical codes $C_1 = [n, k_1, d_1]_p$ and $C_2 = [n, k_2, d_2]_p$ over F_p such that $C_1^{\perp} \subseteq C_2$. This translates to the

[6] The reverse of an encoding circuit is a decoding circuit of a quantum code.

TABLE I. Comparison of noise thresholds for the proposed codes against the theoretical upper bound and previously proposed quantum codes for distilling the $|T\rangle$ state in [6] and $|S\rangle$ state in [7].

Comparison of noise thresholds for distilling $\|T\rangle$ and $\|S\rangle$ states over the depolarizing channel					
Magic states	Theoretical threshold limit over the depolarizing channel	Proposed quantum codes	Threshold	State-of-the-art quantum codes	Threshold
$\|T\rangle$	0.5	$[[31, 1, 16]]_2$	0.498	$[[5, 1, 3]]_2$	0.173
$\|S\rangle$	0.75	$[[13, 1, 7]]_3$ and $[[29, 1, 15]]_3$	0.425 and 0.71	$[[11, 1, 5]]_3$	0.387

condition $\mathrm{H}_1\mathrm{H}_2^{\mathrm{T}} = 0 \bmod (p)$, where H_1 and H_2 are the parity check matrices of the classical codes C_1 and C_2, respectively.

A $[[n, k_1 + k_2 - n, d \geqslant \min(d_1, d_2)]]_p$ qudit quantum stabilizer code Q_1 can be prepared from dual-containing classical codes C_1 and C_2 using the CSS construction [1] with stabilizer generator matrix given as

$$\mathrm{H}_{\mathrm{CSS}} = \begin{bmatrix} \mathrm{H}_1 & 0_{(n-k_1)\times n} \\ 0_{(n-k_2)\times n} & \mathrm{H}_2 \end{bmatrix}, \tag{72}$$

where $0_{(n-k_1)\times n}$ and $0_{(n-k_2)\times n}$ are zero matrices. From Eq. (72), the total number of stabilizer generators in the quantum CSS code is $2n - k_1 - k_2$.

Let us consider first $w = k_1 + k_2 - n$ qudits state $|\Psi\rangle$ to be encoded using an encoding map $\mathcal{E}$ and $a = 2n - k_1 - k_2$ ancilla qudits in state $|0\rangle$ such that $\mathcal{E}|\Psi\rangle|0\rangle^{\otimes a} = |\Phi\rangle$, where $|\Phi\rangle$ is a quantum codeword. Since state $|0\rangle$ is stabilized by the Z gate and state $|\Psi\rangle$ is an unknown state stabilized by $I^{\otimes w}$ only, therefore, the n qudit state $|\Psi\rangle|0\rangle^{\otimes a}$ is stabilized by the stabilizers of the form $\tilde{\mathrm{H}} = [0_{(n-w)\times n}|0_{(n-w)\times w}S_{(n-w)\times(n-w)}]$, where S is any general matrix containing elements from $\mathbb{F}_p$. These are the only stabilizer generators for the initial n qudit state [12].

To obtain an encoding operator, first, we apply Gaussian elimination with row operations and column swappings to simplify the stabilizer matrix $\mathrm{H}_{\mathrm{CSS}}$ into row echelon form as $\tilde{\mathrm{H}}_{\mathrm{CSS}} = [\begin{smallmatrix} \mathrm{H}^{(x)} & 0_1 \\ 0_2 & \mathrm{H}^{(z)} \end{smallmatrix}]$, where 0_1 and 0_2 are the zero matrices,

$$\mathrm{H}^{(x)} = [\mathrm{I}^{(x)}_{\rho_1} | \mathrm{A}^{(x)}_{\rho_1\times\rho_2} | \mathrm{B}^{(x)}_{\rho_1\times(k_1+k_2-n)}], \tag{73}$$

$$\mathrm{H}^{(z)} = [\mathrm{I}^{(z)}_{\rho_2} | \mathrm{A}^{(z)}_{\rho_2\times\rho_1} | \mathrm{B}^{(z)}_{\rho_2\times(k_1+k_2-n)}], \tag{74}$$

$\rho_1 = n - k_1$, and $\rho_2 = n - k_2$.

We know that, for CSS code, the (i, j)th entry of the matrix S such that $S = \mathrm{H}^{(x)}\mathrm{H}^{(z)^{\mathrm{T}}}$ is given by[7]

$$\begin{aligned} S_{i,j} &= \mathrm{I}^{(x)}_{i,*}\mathrm{I}^{(z)}_{*,j} + \mathrm{A}^{(x)}_{i,*}\mathrm{A}^{(z)}_{*,j} + \mathrm{B}^{(x)}_{i,*}\mathrm{B}^{(z)}_{*,j}, \\ &= \delta_{i,j} + \sum_{k\in S_p} k\, M^{(i,j)}_k \equiv 0 \bmod (p)\,, \end{aligned} \tag{75}$$

where $S_p = \{1, 2, \ldots, p-1\}$ and

$$M^{(i,j)}_k = \sum_{l:\mathrm{A}^{(x)}_{i,l}=k} \mathrm{A}^{(z)}_{j,l} + \sum_{l:\mathrm{B}^{(x)}_{i,l}=k} \mathrm{B}^{(z)}_{j,l}.$$

On solving Eq. (75), we obtain

$$\sum_{k\in S_p} k\, M^{(i,j)}_k = \begin{cases} (p-1) \bmod (p), & i = j \\ 0 \bmod (p), & i \neq j. \end{cases} \tag{76}$$

The objective is to remove all the X stabilizers to obtain $\tilde{\mathrm{H}}$ from $\tilde{\mathrm{H}}_{\mathrm{CSS}}$. Therefore, on iterating over the rows of $\mathrm{H}^{(x)}$, we can make all the nonzero elements of the submatrices $\mathrm{A}^{(x)}$, $\mathrm{B}^{(x)}$ zero using multiplication (MUL) and ADD gates.

For an ith row r_i of $\mathrm{H}^{(x)}$, the elements of the submatrices $\mathrm{A}^{(x)}$ and $\mathrm{B}^{(x)}$ are converted to $p-1$ using $M^{(j)}_{\gamma_j}$ gates such that $\gamma_j^{-1}\mathrm{H}^{(x)}_{i,j} = p - 1, \forall\, j \in V = \{\rho_1 + 1, \ldots, n\}$. Next, on applying $\mathrm{ADD}^{(i,j)}_p$, $\forall\, j \in V$ gates, the elements of the r_i row become zero for all columns $j \in V$ of $\mathrm{H}^{(x)}$ as element 1 present at the (i, i) index in $\mathrm{I}^{(x)}$ will be added to all the columns indexed in V where elements are $p-1$ due to MUL gates used [12]. We store the MUL and ADD gates used for the ith row in U_i.

The effect of U_i on the $\mathrm{H}^{(z)}$ matrix will make (i, i)th element of $\mathrm{I}^{(z)}$ submatrix zero. The MUL gates used in U_i will multiply γ_j to the columns present in V of $\mathrm{H}^{(z)}_{i,*}$ such that

$$\gamma_j \equiv \mathrm{H}^{(x)}_{i,j}(p-1) \bmod (p), \;\; \forall\, j \in V, \tag{77}$$

as given in Table II.

Since ADD gates affect the control qudits in Z stabilizers as shown in Table II, ADD gates in U_i will change the elements of $\mathrm{I}^{(z)}$ submatrix, i.e., the columns of the identity submatrix $\mathrm{I}^{(z)}$, which are the control qudits columns will be subtracted from[8]

$$W_{i,j} = \sum_{k\in S_p} \gamma_k M^{(i,j)}_k. \tag{78}$$

Using Eq. (77), we get $\gamma_k = k(p-1)$, $\forall\, k \in S_p$, and then Eq. (78) becomes

$$W_{i,j} = (p-1) \sum_{k\in S_p} k M^{(i,j)}_k. \tag{79}$$

[7] For $p = 3$, the value of $S_{i,j}$ for $\mathrm{H}^{(x)}_{i,*} = [1\,2\,0\,2\,1]$ and $\mathrm{H}^{(z)}_{j,*} = [2\,2\,1\,2\,2]$ is given by $S_{i,j} = [1\,2\,0\,2\,1][2\,2\,1\,2\,2]^{\mathrm{T}} = 1M^{(i,j)}_1 + 2M^{(i,j)}_2 \equiv 0 \bmod (p)$, where $M^{(i,j)}_1 = M^{(i,j)}_2 = 4$.

[8] For example, consider two orthogonal rows $\underline{h}_x = [1\,0\,1\,1\,2]$ and $\underline{h}_z = [1\,0\,1\,2\,1]$ such that $\underline{h}_x\underline{h}_z^{\mathrm{T}} \equiv 0 \bmod (p)$, for $p = 3$. First, we apply $U_1 = M^{(3)}_2 M^{(4)}_2$ gate that will make $\underline{h}_x = [1\,0\,2\,2\,2]$ and $\underline{h}_z = [1\,0\,2\,1\,1]$ and then apply $U = \mathrm{ADD}^{(1,5)}_3\mathrm{ADD}^{(1,4)}_3\mathrm{ADD}^{(1,3)}_3$ gate such that $\underline{h}_x = [1\,0\,0\,0\,0]$ and the effect of U on $\underline{h}_z$ will subtract $W = 4 \equiv 1 \bmod (p)$, which is the sum of last three entries of $\underline{h}_z$ after application of U_1 will change $\underline{h}_z$ as $\underline{h}_z = [0\,0\,2\,1\,1]$.

TABLE II. Description of basic Clifford gates and transformation of the vectors isomorphic to the Pauli group. For example, $[S\ T|U\ V] \cong X^S Z^U \otimes X^T Z^V$.

Basic Clifford gates			
Gate	Symbol	Operation	Transformation of vector
ADD gate	$\mathrm{ADD}_p^{(i,j)}$	$\sum_{a,b\in\mathbb{F}_p} \|a\rangle\langle a\|_i \otimes \|a+b\rangle\langle b\|_j$	$[S\ T\|U\ V] \rightarrow [S\ T+S\|U-V\ V]$
Multiplication gate	$\mathrm{M}_\alpha^{(i)}$	$\sum_{b\in\mathbb{F}_p} \|b\rangle\langle \alpha^{-1}b\|_i$ for $\alpha \in \mathbb{F}_p\backslash\{0\}$	$[S\ \|T] \rightarrow [\alpha^{-1}S\ \|\alpha T]$
DFT gate	$\mathrm{DFT}_p^{(i)}$	$\frac{1}{\sqrt{p}} \sum_{a,b\in\mathbb{F}_p} \omega^{ab}\|a\rangle\langle b\|_i$	$[S\ \|T] \rightarrow [T\ \|-S]$

Using Eq. (76), we get

$$W_{i,j} = \begin{cases}(p-1)^2 \equiv 1 \bmod (p), & i=j\\ 0 \bmod (p), & i \neq j.\end{cases} \tag{80}$$

As $W_{i,j}$ is 1 for $i=j$ so it will make (i,i)th element of $\mathrm{I}^{(z)}$ zero when ith column of $\mathrm{I}^{(z)}$ is a control column. Therefore, the ith column of $\mathrm{I}^{(z)}$ will become zero due to ADD gates of U_i. Hence, for CSS codes, if the operator U_i is making elements of the $\mathrm{H}^{(x)}$ matrix zero using MUL and ADD gates, then the elements of the $\mathrm{H}^{(z)}$ matrix will become zero, where elements of the $\mathrm{H}^{(x)}$ and $\mathrm{H}^{(z)}$ matrices are at the target and control positions of the ADD gates used. On repeating this process sequentially for all ρ_1 rows of $\mathrm{H}^{(x)}$, we make submatrices $\mathrm{A}^{(x)}$ and $\mathrm{B}^{(x)}$ zero, and also first ρ_1 columns of the $\mathrm{H}^{(z)}$ matrix zero. We store all the gates used in $U_1 = \prod_{i=1}^{\rho_1} U_i$.

As we have positioned w logical qudits towards the end in the quantum code Q_1, the submatrix $\mathrm{B}_{\rho_2\times w}^{(z)}$ is made zero to get only I operators on the logical qudits as $\mathrm{B}^{(x)}$ submatrix is already zero. To do this, we solve a linear equation $\mathrm{A}^{(z)}X = \mathrm{B}^{(z)}$ to obtain matrix $X_{a\times w}$, where $a = n - w - \rho_2$. The MUL and ADD gates can be placed according to the columns of the matrix X as follows:

For every column $\underline{x}$ of $X_{a\times w}$, $M_{\gamma^{-1}}^{(j+\rho_2)}\mathrm{ADD}_p^{(i+n-w,j+\rho_2)}M_\gamma^{(j+\rho_2)}$ gate, where $i+n-w$ are the positions of the columns of $\mathrm{B}^{(z)}$ and $j+\rho_2$ are the positions of the columns of $\mathrm{A}^{(z)}$ for $i \in \{1,2,\dots,w\}$ and $j \in \{1,2,\dots,a\}$, is applied for the nonzero element γ at the jth index of the column $\underline{x}$ to get the same column as of the submatrix $\mathrm{B}^{(z)}$, using the linear combination of the columns of submatrix $\mathrm{A}^{(z)}$. As the control qudits of the ADD gates are the columns of $\mathrm{B}^{(z)}$, due to the property of ADD gate as shown in Table II, every column of the $\mathrm{B}^{(z)}$ submatrix can be made zero. U_2 is the product of MUL and ADD gates generated from the matrix X.

Finally, to obtain an encoding operator $\mathcal{E}$, we apply T gate, which is the product of DFT gates, i.e., $T = \prod_{i=1}^{\rho_1} \mathrm{DFT}_p^{(i)}$. The T gate is applied on first ρ_1 qudits to interchange the $\mathrm{I}_{\rho_1}^{(x)}$ submatrix with the first ρ_1 zero columns of 0_1 matrix (zero matrix representing Z stabilizers in the first ρ_1 rows of the $\tilde{\mathrm{H}}_{\mathrm{CSS}}$ matrix) to make $\mathrm{H}^{(x)}$ matrix zero. The decoding operator U is given by $U = TU_2U_1$ and the encoding operator is $\mathcal{E} = U^\dagger$.

We now summarize the procedure to obtain the encoding operator in Algorithm 1. The depth of the quantum encoding operator obtained is $O[\rho_1 + a(n-\rho_1-\rho_2)] = O(\rho_1 + aw)$ as in step 2 only ρ_1 number of MUL and ADD gates are applied and in step 4 MUL and ADD gates are applied up to the dimension of the matrix X.

We compare our encoding procedure with [11] in terms of complexity of the encoder. In step 2 of the encoding process, we apply ADD gates to make $[\mathrm{A}^{(x)}|\mathrm{B}^{(x)}]$ matrix zero, having a size $\rho_1 \times k_1$, so, $\rho_1 k_1$ number of ADD gates are required. In step 3 of the encoding process, we make $\mathrm{B}_{\rho_2\times w}^{(z)}$ matrix zero, requiring w ADD gates targeted to the ρ_1 columns of the $\mathrm{A}_{\rho_2\times\rho_1}^{(z)}$ matrix, so, $w\rho_1$ two-qudit ADD gates are required. The total number of two-qudit ADD gates in our encoder is $\rho_1(k_1+w)$, which is less compared to the $k_1 n - \binom{k_1+1}{2} + (k_2-k_1)\rho_2$[9] in [11]. The reader must note that the efficient encoding procedures described in [12] are applicable for generic qudit non-CSS entanglement unassisted and entanglement assisted quantum codes. Since we can exploit the structure of the CSS construction, the proposed encoders in this section are compared to the work in [11].

The reader must note that the depth of the quantum encoding operator increases with increasing number of logical qudits of the code, i.e., w, and does not depend on the

ALGORITHM 1. Algorithm to obtain an encoding circuit of quantum CSS code.

```
Input: Stabilizer check matrix $\mathrm{H}_{\mathrm{CSS}}$.
Output: Encoding operator $U$.
Outline of the steps:
(1) Apply Gaussian elimination on the matrix $\mathrm{H}_{\mathrm{CSS}}$ to convert to
row echelon form $\tilde{\mathrm{H}}_{\mathrm{CSS}} = [\begin{smallmatrix}\mathrm{H}^{(x)} & 0\\ 0 & \mathrm{H}^{(z)}\end{smallmatrix}]$.
(2) For each nonzero row $i$ of $\mathrm{H}^{(x)}$ do
        For each column $j = \mathrm{rank}(\mathrm{H}^{(x)}) + 1$ to $n$ do
            If $\mathrm{H}_{i,j}^{(x)}$ is nonzero then do
                Find an operator $M_\gamma^{(j)}$ such that $\gamma^{-1}\mathrm{H}_{i,j}^{(x)} = p-1$.
                Store $U_1 \leftarrow \mathrm{ADD}_p^{(i,j)}M_\gamma^{(j)}$ and $T \leftarrow \mathrm{DFT}_p^{(i)}$ to make
                $\mathrm{H}^{(x)}$ matrix zero.
(3) Solve for the matrix $X$ using equation $\mathrm{A}^{(z)}X = \mathrm{B}^{(z)}$.
(4) For each nonzero column $i$ of $X$ do
        For each row $j$ of $X$ check
            If $X_{j,i}$ is nonzero then do
                Store $U_2 \leftarrow M_{X_{j,i}^{-1}}^{(\rho_2+j)}\mathrm{ADD}_p^{(i+n-w,j+\rho_2)}M_{X_{j,i}}^{(\rho_2+j)}$ to make
                $\mathrm{B}^{(z)}$ matrix zero.
(5) Assign $U \leftarrow TU_2U_1$
(6) return $U^\dagger$
```

[9] Proposition 1 in [11] has only $k_1 n - \binom{k_1+1}{2}$ number of ADD gates. However, there are extra $(k_2-k_1)\rho_2$ ADD gates that were not considered, but clearly indicated in the example of Sec. 4 in [11] with 16 ADD gates.

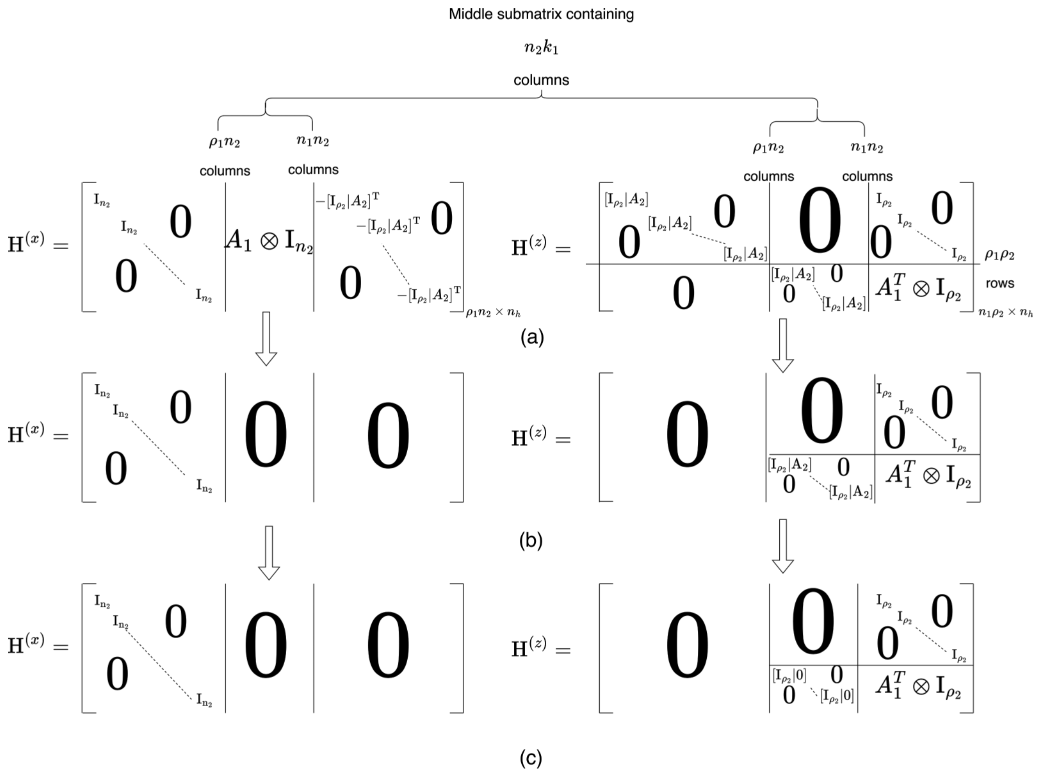

Middle submatrix containing
n_2k_1
columns
$\rho_1 n_2$
columns
n_1n_2
columns
$\mathrm{H}^{(x)} =$
$A_1 \otimes \mathrm{I}_{n_2}$
$-[\mathrm{I}_{\rho_2}|A_2]^T$
$\rho_1 n_2 \times n_h$
$\mathrm{H}^{(z)} =$
$[\mathrm{I}_{\rho_2}|A_2]$
$A_1^T \otimes \mathrm{I}_{\rho_2}$
$\rho_1\rho_2$
rows
$n_1\rho_2 \times n_h$
$[\mathrm{I}_{\rho_2}|0]$
(a)
(b)
(c)

FIG. 10. The transformation of stabilizer matrices $\mathrm{H}^{(x)}$ and $\mathrm{H}^{(z)}$ is shown along the steps of Algorithm 2. (a), (b) Show the transformation during step 2 of the Algorithm 2, where MUL and ADD gates are applied with respect to $\mathrm{H}^{(x)}$ matrix to make the second and third submatrices of $\mathrm{H}^{(x)}$ zero, and the corresponding effect on $\mathrm{H}^{(z)}$ matrix is also shown. (b), (c) Show the transformation during step 3 of Algorithm 2, where MUL and ADD gates are applied to the block matrix A to make it zero in the second submatrix of $\mathrm{H}^{(z)}$, and the second submatrix of $\mathrm{H}^{(x)}$ matrix is already zero.

length n of the quantum CSS code. We provide an example of $[[8,4,3]]_3$ CSS code encoder using Algorithm 1 in Appendix B.

B. Encoding of nondual containing codes using hypergraph product codes

Consider two classical codes C_1 and C_2 with parameters $[n_1,k_1,d_1]_p$ and $[n_2,k_2,d_2]_p$ with parity check matrices $\mathrm{H}_{\rho_1\times n_1}$ and $\tilde{\mathrm{H}}^{\mathrm{T}}_{\rho_2\times n_2}$, respectively, such that $\mathrm{H}\tilde{\mathrm{H}}^{\mathrm{T}} \neq 0 \bmod (p)$. This implies $C_1^\dagger \nsubseteq C_2$ or $C_2^\dagger \nsubseteq C_1$. Therefore, with CSS construction, quantum code using C_1 and C_2 cannot be generated. The hypergraph construction [13] can be used to generate a $[[n_h,k_h,d_h]]_p$, where $n_h = n_1n_2 + \rho_1\rho_2$, $k_h = k_1k_2$, and $d_h = \min(d_1,d_2,\tilde{d}_1,\tilde{d}_2)$ quantum code using two nondual containing classical codes, where $\tilde{d}_1$ and $\tilde{d}_2$ are the code distances of the transposed codes C_1^{T} and C_2^{T} with parity check matrices H^{T} and $\tilde{\mathrm{H}}^{\mathrm{T}}$.

The stabilizer generator matrix of the hypergraph product codes is in the CSS form as in Eq. (72):

$$\mathrm{H}_{\mathrm{CSS}} = \begin{bmatrix} \mathrm{H}^{(x)} & 0_1 \\ 0_2 & \mathrm{H}^{(z)} \end{bmatrix}, \tag{81}$$

where

$$\mathrm{H}^{(x)} = \left[\mathrm{H}\otimes \mathrm{I}_{n_2}\middle| -\mathrm{I}_{\rho_1}\otimes\tilde{\mathrm{H}}^{\mathrm{T}}\right], \tag{82}$$

$$\mathrm{H}^{(z)} = \left[\mathrm{I}_{n_1}\otimes\tilde{\mathrm{H}}\middle|\mathrm{H}^{\mathrm{T}}\otimes \mathrm{I}_{\rho_2}\right], \tag{83}$$

and 0_1, 0_2 are zero matrices of dimension $\rho_1 n_2\times n_h$ and $\rho_2 n_1 \times n_h$ such that $\mathrm{H}^{(x)}\mathrm{H}^{(z)^{\mathrm{T}}} \equiv 0 \bmod (p)$ [13]. The encoding procedure resembles the procedure discussed in Sec. IV C. However, the number of gates increases because hypergraph construction has $n_1n_2 + \rho_1\rho_2$ number of physical qudits. Although hypergraph construction provides quantum codes with large code length and has short cycles on the Tanner graph, it can retain the sparseness of the classical LDPC codes, which is not in the case of generalized Shor code [13]. The reader must note that entanglement assisted (EA) codes [12] can generate quantum codes from two nondual containing classical codes and the code is devoid of short cycles. However, EA codes have to maintain noise-free entangled states, which is not required in the case of hypergraph construction.

To obtain an encoding operator, first, we apply Gaussian elimination on the matrices H and $\tilde{\mathrm{H}}$ and obtain matrices in row echelon form as $\mathrm{H}_{\rho_1\times n_1} = [\mathrm{I}_{\rho_1}|\mathrm{A}_{1_{\rho_1\times(n_1-\rho_1)}}]$ and $\tilde{\mathrm{H}}_{\rho_2\times n_2} = [\mathrm{I}_{\rho_2}|\mathrm{A}_{2_{\rho_2\times(n_2-\rho_2)}}]$. The corresponding stabilizer matrices $\mathrm{H}^{(x)}$ and $\mathrm{H}^{(z)}$ in Eqs. (82) and (83) become

$$\mathrm{H}^{(x)} = \left[\mathrm{I}_{\rho_1}\otimes \mathrm{I}_{n_2}\middle|\mathrm{A}_1\otimes\mathrm{I}_{n_2}\middle| -\mathrm{I}_{\rho_1}\otimes\tilde{\mathrm{H}}^{\mathrm{T}}\right], \tag{84}$$

$$\mathrm{H}^{(z)} = \left[\begin{array}{c|c} \mathrm{I}_{n_1}\otimes\tilde{\mathrm{H}} & \begin{array}{c}\mathrm{I}_{\rho_1}\otimes\mathrm{I}_{\rho_2}\\ \hline \mathrm{A}_t\otimes\mathrm{I}_{\rho_2}\end{array}\end{array}\right], \tag{85}$$

where $\mathrm{A}_t = \mathrm{A}_1^{\mathrm{T}}$. Equations (84) and (85) in expanded form are shown in Fig. 10(a).

ALGORITHM 2. Algorithm to obtain an encoding circuit of quantum hypergraph code.

```
Input: Classical parity check matrices H and $\tilde{\mathrm{H}}$.
Output: Encoding operator $U$.
Outline of the steps:
(1) Apply Gaussian elimination on the matrices H and $\tilde{\mathrm{H}}$ to
convert the matrices into row echelon form to generate stabilizer
matrices $\mathrm{H}^{(x)}$ and $\mathrm{H}^{(z)}$ using Eqs. (82) and (83).
(2) For each nonzero row $i$ of $\mathrm{H}^{(x)}$ do
        For each column $j$ from $\mathrm{rank}(\mathrm{H}^{(x)})+1$ to $n_h$ do
            If $\mathrm{H}^{(x)}_{i,j}$ is nonzero then do
                Find an operator $M^{(j)}_{\gamma}$ such that $\gamma^{-1}\mathrm{H}^{(x)}_{i,j}=p-1$,
                Store $W_1 \leftarrow \mathrm{ADD}^{(i,j)}_p M^{(j)}_{\gamma}$ to make the first
                submatrix of $\mathrm{H}^{(x)}$ zero.
(3) For each row $i$ of $\mathrm{H}^{(z)}$ from $\rho_1\rho_2+1$ to $n_1\rho_2$ do
    Assign $u \leftarrow \lfloor \frac{i-\rho_1\rho_2}{\rho_2} \rfloor$.
        For each column $j$ of $\mathrm{H}^{(z)}$ from $n_2\rho_1+\rho_2+un_2$ to
          $n_1\rho_2+(u+1)n_2$
            If $\mathrm{H}^{(z)}_{i,j}$ is nonzero then
                Find an operator $M^{(j)}_{\gamma}$ such that $\gamma\mathrm{H}^{(z)}_{i,j}=p-1$,
                Store $W_2 \leftarrow \mathrm{ADD}^{(n_2\rho_1+uk_2+i-\rho_1\rho_2,j)}_p M^{(j)}_{\gamma}$ to make the
                middle submatrix of $\mathrm{H}^{(z)}$ zero.
(4) Assign $U \leftarrow TW_2W_1$. The operator $T \leftarrow \prod_{i=1}^{\rho_1 n_2} \mathrm{DFT}^{(i)}_p$
    interchanges the first submatrix of $\mathrm{H}^{(x)}$
    with the first zero submatrix of $\mathrm{H}^{(z)}$.
(5) return $U^{\dagger}$.
```

To make $\mathrm{H}^{(x)}$ matrix zero, we apply MUL and ADD gates in accordance with the $[\mathrm{A}_1 \otimes \mathrm{I}_{n_2}| - \mathrm{I}_{\rho_1} \otimes \tilde{\mathrm{H}}^{\mathrm{T}}]$ submatrix and make it zero. These gates will also make the first $n_2\rho_1$ columns of the $\mathrm{H}^{(z)}$ matrix zero since $\mathrm{H}^{(x)}\mathrm{H}^{(z)^{\mathrm{T}}} \equiv 0 \bmod (p)$. Application of MUL and ADD gates will make target columns of the $\mathrm{H}^{(x)}$ matrix zero as well as control columns of the $\mathrm{H}^{(z)}$ matrix zero as shown in Fig. 10(b). This is similar to how the reduction operations are done for dual containing codes in Sec. V A. We store the MUL and ADD gates in U_1.

Now, we will make any $k_h = k_1k_2$ nonzero columns of the $\mathrm{H}^{(z)}$ matrix zero for the logical qudits, similar to the procedure in Sec. V A. The middle submatrix of $\mathrm{H}^{(z)}$ has first $\rho_1\rho_2$ rows zero, and the rest ρ_2k_1 rows have $\tilde{\mathrm{H}} = [\mathrm{I}_{\rho_2}|\mathrm{A}_2]$ in the diagonal of the middle submatrix of $\mathrm{H}^{(z)}$ as shown in Fig. 10(a). The A_2 submatrix of every block-diagonal matrix (BDM) $\tilde{\mathrm{H}}$ can be made zero using MUL and ADD gates controlled from the columns of the I_{ρ_2} submatrix of each $\tilde{\mathrm{H}}$, as shown in Fig. 10(c). As there are k_1 BDMs in the middle submatrix of $\mathrm{H}^{(z)}$ and A_2 submatrix of every $\tilde{\mathrm{H}}$ has k_2 number of columns, so total number of generated zero columns in the middle submatrix of $\mathrm{H}^{(z)}$ are $k_1k_2 = k_h$. We store the MUL and ADD gates for making k_h columns of $\mathrm{H}^{(z)}$ zero in U_2.

Finally, to make nonzero submatrix $[\mathrm{I}_{\rho_1} \otimes \mathrm{I}_{n_2}]$ of $\mathrm{H}^{(x)}$ zero, we apply $T = \prod_{i=1}^{\rho_1 n_2} \mathrm{DFT}^{(i)}_p$ gate to interchange the submatrix $[\mathrm{I}_{\rho_1} \otimes \mathrm{I}_{n_2}]$ of $\mathrm{H}^{(x)}$ with the first $\rho_1 n_2$ columns of zero matrix 0_1, where 0_1 is a zero matrix as given in Eq. (81). We get the decoding operator for hypergraph product code as $D = T\,U_2\,U_1$. The encoding operator is simply $\mathcal{E} = U^{\dagger}$.

In Algorithm 2, we provide an algorithmic framework to obtain an encoding operator of the hypergraph product code. The depth of the quantum encoding operator obtained is $O(\rho_1 n_2 + k_1\rho_2)$ since in steps 2 and 3 $\rho_1 n_2$ and $k_1\rho_2$ MUL and ADD gates are applied.

We provide examples of hypergraph product code encoder using Algorithm 2 in Appendix B.

The reader must note that the celebrated quasicyclic low-density parity check (LDPC) codes can be derived from lifted product (LP) codes that generalize bicycle codes [25] and hypergraph product codes [13], introduced in [26]. These codes have nonzero asymptotic rates and almost linear distance with the scaling of code length, making them suitable for practical purposes. Such QC codes are widely used in classical communication systems, such as in wireless systems [27] and data storage devices [28,29] (HDDs and SSDs). LP codes are just the lifted versions of hypergraph product codes. Since efficient quantum codes can be from this broad family of LDPC codes, we believe Algorithm 2 can be used for building encoders for the family of LP codes as well.

VI. CONCLUSIONS AND FUTURE WORK

We developed a geometric framework for magic-state distillation and provided an exact analysis for the noise threshold in terms of the noise over the magic states and qudit stabilizer code parameters. Through information-theoretic analysis, we proved the convergence analysis of the MSD protocol. We provided scaling results that achieve the noise threshold in terms of the quantum code minimum distance and code length. We provided 31-qubit BCH code for distilling $|T\rangle$ and 29-qutrit code for distilling $|S\rangle$ states with the noise thresholds of 0.498 and 0.71, respectively, over the quantum depolarizing channel. These codes are derived based on the scaling laws for the code parameters towards reaching the theoretical limits for the noise threshold. Finally, we proposed two encoding algorithms for CSS and non-CSS qudit quantum stabilizer codes. The proposed encoding architectures and quantum codes are useful for designing fault-tolerant quantum circuits for computing and communication applications.

Going forward, it would be interesting to develop fault-tolerant circuits based on QECCs and magic-state distillation for practical quantum neural networks, mindful of the constraints on decoherence and quantum circuit complexity.

ACKNOWLEDGMENTS

This work is financially supported by the Department of Science and Technology (DST), Govt. of India, with Ref No. SERB/F/3132/2023-2024.

APPENDIX A: IMPORTANT PROOFS

In this Appendix, we have provided the proofs of all the three theorems used in the main text.

We first prove the Theorem 1.

Proof. First, we calculate the entropy of the input state at the tth iteration of MSD as $H(\sigma)^{(t)}_{\mathrm{in}} = -\mathrm{tr}[\sigma^{(t)}_{\mathrm{in}} \log_2(\sigma^{(t)}_{\mathrm{in}})]$, where

$$\sigma^{(t)}_{\mathrm{in}} = (\sigma^{(t)})^{\otimes n} = \sum_{\underline{z} \in \mathbb{F}_p^n} w^{(t)}_{\underline{z}} |\xi_{\underline{z}}\rangle\langle\xi_{\underline{z}}|. \tag{A1}$$

We can simplify $H(\sigma)_{\text{in}}^{(t)}$ as

$$H(\sigma)_{\text{in}}^{(t)}$$

$$= -\text{tr}\left[\sum_{\underline{z}\in\mathbb{F}_p^n} w_{\underline{z}}^{(t)}|\xi_{\underline{z}}\rangle\langle\xi_{\underline{z}}|\log_2\left(\sum_{\underline{y}\in\mathbb{F}_p^n} w_{\underline{y}}^{(t)}|\xi_{\underline{y}}\rangle\langle\xi_{\underline{y}}|\right)\right], \tag{A2}$$

$$H(\sigma)_{\text{in}}^{(t)} = -\sum_{\underline{z},\underline{y}\in\mathbb{F}_p^n} w_{\underline{z}}^{(t)}\log_2\left(w_{\underline{y}}^{(t)}\right)\langle\xi_{\underline{z}}||\xi_{\underline{y}}\rangle\text{tr}(|\xi_{\underline{z}}\rangle\langle\xi_{\underline{y}}|). \tag{A3}$$

Since $|\xi_{\underline{z}}\rangle$'s are orthonormal states, and $\text{tr}(|\xi_{\underline{z}}\rangle\langle\xi_{\underline{z}}|) = 1$, we have

$$H(\sigma)_{\text{in}}^{(t)} = -\sum_{\underline{z}\in\mathbb{F}_p^n} w_{\underline{z}}^{(t)}\log_2\left(w_{\underline{z}}^{(t)}\right) = H(w^{(t)}), \tag{A4}$$

where $H(w^{(t)})$ is classical Shannon entropy.

The quantum entropy of the output state $\sigma_{\text{out}}^{(t)} = \frac{\text{P}\sigma_{\text{in}}^{(t)}\text{P}}{P_s^{(t)}}$ of the MSD procedure at time t is given by

$$H(\sigma)_{\text{out}}^{(t)} = -\text{tr}\left[\frac{\text{P}(\sigma^{(t)})^{\otimes n}\text{P}}{P_s^{(t)}}\log_2\left(\frac{\text{P}(\sigma^{(t)})^{\otimes n}\text{P}}{P_s^{(t)}}\right)\right] \tag{A5}$$

$$= -\text{tr}\left(\frac{\text{P}(\sigma^{(t)})^{\otimes n}\text{P}}{P_s^{(t)}}2\,\log_2(\text{P})\right)$$
$$-\,\text{tr}\left(\frac{\text{P}(\sigma^{(t)})^{\otimes n}\text{P}}{P_s^{(t)}}\log_2((\sigma^{(t)})^{\otimes n})\right)$$
$$+\,\text{tr}\left(\frac{\text{P}(\sigma^{(t)})^{\otimes n}\text{P}}{P_s^{(t)}}\log_2\left(P_s^{(t)}\right)\right), \tag{A6}$$

where $P_s^{(t)} = \text{tr}(\text{P}\sigma_{\text{out}}^{(t)})$, i.e., the syndrome probability.

Since the projection operator has unit eigenvalues, we have $\log_2(\text{P})$=0. Therefore, $H(\sigma)_{\text{out}}^{(t)}$ can be simplified as

$$H(\sigma)_{\text{out}}^{(t)} = -\frac{1}{P_s^{(t)}}\text{tr}\{\text{P}(\sigma^{(t)})^{\otimes n}\text{P}\,\log_2[(\sigma^{(t)})^{\otimes n}]\}$$
$$+\frac{\log_2\left(P_s^{(t)}\right)}{P_s^{(t)}}\text{tr}[\text{P}(\sigma^{(t)})^{\otimes n}\text{P}]. \tag{A7}$$

We simplify Eq. (A7) as follows:

$$H(\sigma)_{\text{out}}^{(t)} = H_1^{(t)} + H_2^{(t)}, \tag{A8}$$

where $H_1^{(t)} = -\frac{1}{P_s^{(t)}}\text{tr}\{\text{P}(\sigma^{(t)})^{\otimes n}\text{P}\,\log_2[(\sigma^{(t)})^{\otimes n}]\}$ and $H_2^{(t)} = \frac{\log_2(P_s^{(t)})}{P_s^{(t)}}\text{tr}[\text{P}(\sigma^{(t)})^{\otimes n}\text{P}]$. Using Eq. (A1), we solve $H_1^{(t)}$ first as follows:

$$H_1^{(t)} = -\frac{1}{P_s^{(t)}}\text{tr}\left[\sum_{i,j\in\mathbb{F}_p}|\phi_i\rangle\langle\phi_i|\left(\sum_{\underline{k}\in\mathbb{F}_p^n} w_{\underline{k}}^{(t)}|\xi_{\underline{k}}\rangle\langle\xi_{\underline{k}}|\right)\right.$$
$$\left.\times|\phi_j\rangle\langle\phi_j|\log_2\left(\sum_{\underline{l}\in\mathbb{F}_p^n} w_{\underline{l}}^{(t)}|\xi_{\underline{l}}\rangle\langle\xi_{\underline{l}}|\right)\right], \tag{A9}$$

$$H_1^{(t)} = -\frac{1}{P_s^{(t)}}\sum_{i,j\in\mathbb{F}_p}\sum_{\underline{k},\underline{l}\in\mathbb{F}_p^n} w_{\underline{k}}^{(t)}\log_2\left(w_{\underline{l}}^{(t)}\right)\langle\phi_i||\xi_{\underline{k}}\rangle\langle\xi_{\underline{k}}||\phi_j\rangle$$
$$\times\,\langle\phi_j||\xi_{\underline{l}}\rangle\text{tr}(|\phi_i\rangle\langle\xi_{\underline{l}}|). \tag{A10}$$

On simplifying Eq. (A10) further, we get

$$H_1^{(t)} = -\frac{1}{P_s^{(t)}}\sum_{\underline{k},\underline{l}\in\mathbb{F}_p^n} w_{\underline{k}}^{(t)}\log_2\left(w_{\underline{l}}^{(t)}\right)\sum_{i\in\mathbb{F}_p}\langle\phi_i||\xi_{\underline{k}}\rangle\langle\xi_{\underline{l}}||\phi_i\rangle$$
$$\times\sum_{j\in\mathbb{F}_p}\langle\xi_{\underline{k}}||\phi_j\rangle\langle\phi_j||\xi_{\underline{l}}\rangle. \tag{A11}$$

For a projection operator P, we get

$$\sum_{j\in\mathbb{F}_p}\langle\xi_{\underline{k}}||\phi_j\rangle\langle\phi_j||\xi_{\underline{l}}\rangle = \langle\xi_{\underline{k}}|\text{P}|\xi_{\underline{l}}\rangle. \tag{A12}$$

We simplify Eq. (A11) as

$$H_1^{(t)} = -\frac{1}{P_s^{(t)}}\sum_{\underline{k},\underline{l}\in\mathbb{F}_p^n} w_{\underline{k}}^{(t)}\log_2\left(w_{\underline{l}}^{(t)}\right)|\langle\xi_{\underline{k}}|\text{P}|\xi_{\underline{l}}\rangle|^2. \tag{A13}$$

Using Eq. (18) and the fact $|\langle\xi_{\underline{k}}|\text{P}|\xi_{\underline{l}}\rangle|^2 \leqslant 1$, we simplify (A13) as

$$H_1^{(t)} \leqslant -\frac{1}{P_s^{(t)}}\sum_{\underline{k}\in\mathbb{F}_p^n} w_{\underline{k}}^{(t)}\log_2\left(w_{\underline{k}}^{(t)}\right). \tag{A14}$$

Similarly, we solve for $H_2^{(t)}$ in (A8), using (A1) as follows:

$$H_2^{(t)} = \frac{\log_2\left(P_s^{(t)}\right)}{P_s^{(t)}}\sum_{i,j\in\mathbb{F}_p}\sum_{\underline{k}\in\mathbb{F}_p^n} w_{\underline{k}}^{(t)}\langle\phi_i||\xi_{\underline{k}}\rangle\langle\xi_{\underline{k}}||\phi_j\rangle\langle\phi_i||\phi_j\rangle. \tag{A15}$$

As $|\phi_i\rangle$'s are an orthonormal basis, simplifying Eq. (A15), we get

$$H_2^{(t)} = \frac{\log_2\left(P_s^{(t)}\right)}{P_s^{(t)}}\sum_{i\in\mathbb{F}_p}\sum_{\underline{k}\in\mathbb{F}_p^n} w_{\underline{k}}^{(t)}|\langle\xi_{\underline{k}}||\phi_i\rangle|^2. \tag{A16}$$

Using Eq. (A12) and $\sum_{\underline{k}\in\mathbb{F}_p^n} w_{\underline{k}}^{(t)} = 1$, we get

$$H_2^{(t)} \leqslant \frac{\log_2\left(P_s^{(t)}\right)}{P_s^{(t)}}. \tag{A17}$$

Using Eqs. (A14) and (A17), (A8) becomes

$$H(\sigma)_{\text{out}}^{(t)} \leqslant \frac{1}{P_s^{(t)}}\left(\log_2\left(P_s^{(t)}\right) - \sum_{\underline{k}\in\mathbb{F}_p^n} w_{\underline{k}}^{(t)}\log_2\left(w_{\underline{k}}^{(t)}\right)\right). \tag{A18}$$

Using Eq. (A4) in (A18), we get

$$H(\sigma)_{\text{out}}^{(t)} \leqslant \frac{1}{P_s^{(t)}}\left[\log_2\left(P_s^{(t)}\right) + H(\sigma)_{\text{in}}^{(t)}\right]. \tag{A19}$$

As $\log_2(P_s^{(t)}) < 0$, from Eq. (A19), it is evident that MSD is a convergent map of the quantum entropic function. ■

Next, we prove Theorem 2.

Proof. We shall first establish that the syndrome probability obtained by projecting the mixed state over the code projection operator improves over iterations as long as we are below the noise threshold. (a) Using Eqs. (18) and (22), the single-qudit output state after the tth iteration of MSD is given by

$$\rho_{\text{out},t} = \frac{1}{P_s^{(t-1)}} \sum_{i\in\mathbb{F}_p} p_i^{(t)} |\xi_i\rangle\langle\xi_i|, \tag{A20}$$

where $p_i^{(t)}$ are fidelity functions and $P_s^{(t-1)}$ is the probability at the tth iteration. As we are operating below the threshold, i.e., $\exists\, \tilde{i}/\ p_{\tilde{i}}^{(t)} \geqslant p_{i\neq\tilde{i}}^{(t)}$.[10] Over n realizations of ρ_{out}, at the tth iteration,

$$\rho_{\text{out},t}^{\otimes n} = \left(\frac{1}{P_s^{(t-1)}}\right)^n \sum_{\underline{x}\in\mathbb{F}_p^n} p_{\underline{x}}^{(t)} |\xi_{\underline{x}}\rangle\langle\xi_{\underline{x}}|, \tag{A21}$$

where $p_{\underline{x}}^{(t)} = p_{x_1}^{(t)} p_{x_2}^{(t)} \ldots p_{x_n}^{(t)}$. Using Eq. (14), the fidelity functions $p_i^{(t+1)}$ become

$$p_i^{(t+1)} = \sum_{\underline{x}\in\mathbb{F}_p^n} \left(\frac{1}{P^{(t-1)}}\right)^n p_{\underline{x}}^{(t)} |\langle\xi_{\underline{x}}||\phi_i\rangle|^2. \tag{A22}$$

We know that the MSD procedure is purifying and operating below the noise threshold. The fidelity functions at tth and $(t+1)$st iterations should satisfy the following relation for some $\tilde{i}$, $p_{\tilde{i}}^{(2)} \geqslant p_{\tilde{i}}^{(1)}$ and the rest $p_{i\neq\tilde{i}}^{(2)} \leqslant p_{i\neq\tilde{i}}^{(1)}$ when $|\xi_{\tilde{i}}\rangle$ is the state to be distilled. Therefore, using Eq. (22), it is clear that the probability of projection onto the code space at the $(t+1)$st iteration is greater than the tth iteration. This holds for all iterations. (b) We next show that an increment in the probability of projection implies purification using the MSD procedure from Eq. (22) as follows. Consider for some kth iteration that the probability of projection is greater than the $(k-1)$st, then

$$P^{(k)} = \left(\frac{1}{P^{(k-1)}}\right)^n \sum_{i\in\mathbb{F}_p} p_{i,k} \geqslant P^{(k-1)},$$

$$\sum_{i\in\mathbb{F}_p} p_{i,k} \geqslant (P^{(k-1)})^{n+1}, \tag{A23}$$

where $P^{(k-1)}$ is the probability of projection at the $(k-1)$st iteration and $p_{i,k}$ is the fidelity function at the kth iteration. Using Eqs. (22) and (A23), we get

$$\sum_{i\in\mathbb{F}_p} p_{i,k} \geqslant \left(\sum_{i\in\mathbb{F}_p} p_{i,k-1}\right)^{n+1}. \tag{A24}$$

As fidelity functions, $p_{i,k}$ are less than 1, it satisfies

$$\sum_{i\in\mathbb{F}_p} p_{i,k} > p_{\tilde{i},k-1}^{n+1} + \left(\sum_{i\in\mathbb{F}_p\backslash\{\tilde{i}\}} p_{i,k-1}\right)^{n+1}. \tag{A25}$$

Let $f_{\tilde{i},k}$ be noise fidelity function at time k for distilling state $|\xi_{\tilde{i}}\rangle$:

$$f_{\tilde{i},k} = \sum_{i\in\mathbb{F}_p\backslash\{\tilde{i}\}} p_{i,k}. \tag{A26}$$

Substituting $f_{\tilde{i},k}$ from Eq. (A26) into (A25), we get

$$p_{\tilde{i},k} + f_{\tilde{i},k} \geqslant p_{\tilde{i},k-1}^{n+1} + f_{\tilde{i},k-1}^{n+1}. \tag{A27}$$

As we assumed that we are operating below threshold, $p_{\tilde{i},k} \geqslant f_{\tilde{i},k}$. On the contrary, if the kth iteration is adding noise, then $f_{\tilde{i},k-1} \leqslant f_{\tilde{i},k}$ and $p_{\tilde{i},k-1} \geqslant p_{\tilde{i},k}$, implying $f_{\tilde{i},k}^{n+1} \geqslant f_{\tilde{i},k-1}^{n+1}$ and $p_{\tilde{i},k}^{n+1} \leqslant p_{\tilde{i},k-1}^{n+1}$. Therefore,

$$p_{\tilde{i},k}^{n+1} + f_{\tilde{i},k}^{n+1} \leqslant p_{\tilde{i},k-1}^{n+1} + f_{\tilde{i},k-1}^{n+1}. \tag{A28}$$

At the $(k+1)$st iteration from Eq. (A27), we have

$$p_{\tilde{i},k+1} + f_{\tilde{i},k+1} \geqslant p_{\tilde{i},k}^{n+1} + f_{\tilde{i},k}^{n+1}. \tag{A29}$$

As a consequence, Eqs. (A28) and (A29) are contradicting with our initial assumption that the syndrome probability increases with iterations. Hence, the MSD procedure is purifying and no iteration is adding noise. This is valid for any iteration t; hence, the probability of projection onto the code space is a nondecreasing function over MSD iterations. ■

Finally, we prove Theorem 3.

Proof. Equations (33) can be simplified by using the fact $P_s \geqslant 1 - 2\epsilon_{\text{out}}$[11] as follows:

$$\sum_{i\in S_p\backslash\{0\}} \left[\left(\frac{\epsilon}{p-1}\right)^d p_i + (1-p_i)(1-\epsilon)^d f(i,\epsilon)\right]$$
$$> \frac{\epsilon_{\text{out}}}{p-1}(1-2\epsilon_{\text{out}}). \tag{A30}$$

Consider a $[[n_1, 1, d_1]]_p$ qudit quantum stabilizer code Q_1 to distill the $|u\rangle$ state with an encoding operator $\mathcal{E}_1$, which is an eigenstate of an operator W with eigenvalue u_W such that $\mathcal{E}_1(|u\rangle) = |U\rangle$ and $\bar{W}|U\rangle = u_W|U\rangle$, where $\bar{W}$ is a logical operator onto the code space.

The code Q_1 also satisfies the distillation conditions described in Sec. II B with noise threshold $\epsilon_1 = \epsilon_{\text{out}} = \epsilon$ by definition [6] and $\tilde{f}_1(\epsilon_1) = \max_{i\in S_p\backslash\{0\}} f(i,\epsilon_1)$. We call the function $\tilde{f}(\epsilon_1)$ maximum noise function. Equation (A30) for the code Q_1 can be written as

$$\left(\frac{\epsilon_1}{p-1}\right)^{d_1} \tilde{p}_1 + (1-\epsilon_1)^{d_1}(p-1-\tilde{p}_1)\tilde{f}_1(\epsilon_1) > \frac{\epsilon_1(1-2\epsilon_1)}{(p-1)}, \tag{A31}$$

where $\tilde{f}_1(\epsilon) = \sum_{i\in S_1}(\frac{\epsilon_1}{(p-1)(1-\epsilon_1)})^i$ and $\tilde{p}_1 = \sum_{i\in S_p\backslash\{0\}} p_i^{(1)}$. $\mathrm{P}|\xi_{\underline{x}}\rangle$ are the noisy states which are not proportional to magic state $|u\rangle$ if $|\underline{x}|_H \in S_1$ as described in Sec. III A, which implies $\mathrm{P}|\xi_{\underline{x}}\rangle$ are the states proportional to noisy states as $\bar{W}\mathrm{P}|\xi_{\underline{x}}\rangle = c_W\mathrm{P}|\xi_{\underline{x}}\rangle$, $c_W \neq u_W$.

Similarly, consider another $[[n_2, 1, d_2]]_p$ qudit quantum stabilizer code Q_2 satisfying distillable conditions as code Q_1 to distill the $|u\rangle$ state with noise threshold ϵ_2 such that $d_2 > d_1$.

[10] There exists a state that has more fidelity than the rest, which will eventually converge to a pure state.

[11] Consider a state $\rho = \frac{1}{P_s}(\epsilon_1|\psi_1\rangle\langle\psi_1| + \epsilon_2|\psi_2\rangle\langle\psi_2|)$ such that $P_s = \epsilon_1 + \epsilon_2$. The probability P_s is lower bounded by $\epsilon_1 - \epsilon_2$. Similarly, for Eq. (33), P_s is lower bounded by $1 - 2\epsilon_{\text{out}}$.

Therefore, Eq. (A30), for the code Q_2, can be written as

$$\left(\frac{\epsilon_2}{p-1}\right)^{d_2}\tilde{p}_2+(1-\epsilon_2)^{d_2}(p-1-\tilde{p}_2)\tilde{f}_2(\epsilon_2)>\frac{\epsilon_2(1-2\epsilon_2)}{(p-1)}, \tag{A32}$$

where $\tilde{f}_2(\epsilon_2)=\sum_{i\in S_2}(\frac{\epsilon_2}{(p-1)(1-\epsilon_2)})^i$, $\tilde{p}_2=\sum_{i\in S_p\setminus\{0\}}p_i^{(2)}$.

As $d_2>d_1$, this implies $(1-\epsilon)^{d_1}>(1-\epsilon)^{d_2}$ and $\epsilon^{d_1}>\epsilon^{d_2}$ and considering $\tilde{p}_2\leqslant\tilde{p}_1$. Therefore, (A31) and (A32) can be simplified as

$$\left(\frac{\epsilon_1}{p-1}\right)^{d_1}P_1+(1-\epsilon_1)^{d_1}(p-1-\tilde{p}_2)\tilde{f}_1(\epsilon_1)>\frac{\epsilon_1(1-2\epsilon_1)}{(p-1)}, \tag{A33}$$

$$\left(\frac{\epsilon_2}{p-1}\right)^{d_1}\tilde{p}_1+(1-\epsilon_2)^{d_1}(p-1-\tilde{p}_2)\tilde{f}_2(\epsilon_2)>\frac{\epsilon_2(1-2\epsilon_2)}{(p-1)}. \tag{A34}$$

On subtracting (A33) and (A34), we get

$$\begin{aligned}\tilde{p}_1\left(\frac{\epsilon_2}{p-1}\right)^{d_1}-\tilde{p}_1\left(\frac{\epsilon_1}{p-1}\right)^{d_1}+(p-1-\tilde{p}_2)\\ \times[(1-\epsilon_2)^{d_1}\tilde{f}_2(\epsilon_2)-(1-\epsilon_1)^{d_1}\tilde{f}_1(\epsilon_1)]\\ >\frac{\epsilon_2(1-2\epsilon_2)}{(p-1)}-\frac{\epsilon_1(1-2\epsilon_1)}{(p-1)}.\end{aligned} \tag{A35}$$

Similarly, on considering $\tilde{p}_1<\tilde{p}_2$, (A31) and (A32) can be written as

$$\left(\frac{\epsilon_1}{p-1}\right)^{d_1}\tilde{p}_2+(1-\epsilon_1)^{d_1}(p-1-\tilde{p}_1)\tilde{f}_1(\epsilon_1)>\frac{\epsilon_1(1-2\epsilon_1)}{(p-1)}, \tag{A36}$$

$$\left(\frac{\epsilon_2}{p-1}\right)^{d_1}\tilde{p}_2+(1-\epsilon_2)^{d_1}(p-1-\tilde{p}_1)\tilde{f}_2(\epsilon_2)>\frac{\epsilon_2(1-2\epsilon_2)}{(p-1)}. \tag{A37}$$

Subtraction of (A36) and (A37) will get us

$$\begin{aligned}\tilde{p}_2\left(\frac{\epsilon_2}{p-1}\right)^{d_1}-\tilde{p}_2\left(\frac{\epsilon_1}{p-1}\right)^{d_1}+(p-1-\tilde{p}_1)\\ \times[(1-\epsilon_2)^{d_1}\tilde{f}_2(\epsilon_2)-(1-\epsilon_1)^{d_1}\tilde{f}_1(\epsilon_1)]\\ >\frac{\epsilon_2(1-2\epsilon_2)}{(p-1)}-\frac{\epsilon_1(1-2\epsilon_1)}{(p-1)}.\end{aligned} \tag{A38}$$

Finally, on subtracting (A38) from (A35), we get

$$\begin{aligned}(\tilde{p}_2-\tilde{p}_1)\left[\left(\frac{\epsilon_2}{p-1}\right)^{d_1}-\left(\frac{\epsilon_1}{p-1}\right)^{d_1}\right]+(\tilde{p}_2-\tilde{p}_1)\\ \times[(1-\epsilon_2)^{d_1}\tilde{f}_2(\epsilon_2)-(1-\epsilon_1)^{d_1}\tilde{f}_1(\epsilon_1)]>0.\end{aligned} \tag{A39}$$

On simplifying (A39), we get

$$\begin{aligned}\left(\frac{\epsilon_2}{p-1}\right)^{d_1}-\left(\frac{\epsilon_1}{p-1}\right)^{d_1}\\ >(1-\epsilon_1)^{d_1}\tilde{f}_1(\epsilon_1)-(1-\epsilon_2)^{d_1}\tilde{f}_2(\epsilon_2).\end{aligned} \tag{A40}$$

If the code Q_2 provides better noise suppression as compared to code Q_1, then the maximum noise functions of codes Q_1 and Q_2 are related as $\tilde{f}_2(\epsilon_2)\leqslant\tilde{f}_1(\epsilon_1)$; therefore, the right-hand side in Eq. (A40) is non-negative and so from the left-hand side, we get $\epsilon_2>\epsilon_1$. As $\epsilon_2>\epsilon_1$ implies $(1-\epsilon_2)^{d_1}<(1-\epsilon_1)^{d_1}$ so Eq. (A40) becomes

$$\begin{aligned}\left[\left(\frac{\epsilon_2}{p-1}\right)^{d_1}-\left(\frac{\epsilon_1}{p-1}\right)^{d_1}\right]\\ >[(1-\epsilon_1)]^{d_1}[\tilde{f}_1(\epsilon_1)-\tilde{f}_2(\epsilon_2)].\end{aligned} \tag{A41}$$

On simplifying Eq. (A41), we get

$$\epsilon_2>(p-1)\left[\left(\frac{\epsilon_1}{p-1}\right)^{d_1}+(1-\epsilon_1)^{d_1}[\tilde{f}_1(\epsilon_1)-\tilde{f}_2(\epsilon_2)]\right]^{\frac{1}{d_1}}. \tag{A42}$$

■

APPENDIX B: EXAMPLES OF ENCODING OPERATORS

We provide illustrative encoding examples of CSS and hypergraph product codes using Algorithms 1 and 2 for shorter length codes due to space limitations. The details of the stabilizers and encoding operators for the $[[31, 1, 16]]_2$ quantum BCH code and $[[29, 1, 15]]_3$ qutrit code that achieve near-threshold magic-state distillation are available in Appendix C.

1. Encoding example of CSS code

Consider a $[8, 6, 3]_3$ classical ternary code with parity check matrix in row echelon form using Gaussian elimination given by $\mathrm{H}^{(z)}=\mathrm{H}^{(x)}=\mathrm{H}=[\begin{smallmatrix}1&0&2&0&1&2&1&2\\0&1&0&2&1&2&2&1\end{smallmatrix}]$ such that $\mathrm{HH}^{\mathrm{T}}\equiv 0\bmod(3)$.

Using the CSS construction, a $[[8, 4, 3]]_3$ quantum code can be generated whose encoding operator is given by Algorithm 1 as follows: On executing step 2 of Algorithm 1, the fifth and seventh columns of the matrix $\mathrm{H}^{(x)}$ do not have the value 2 in the first row, so the required gate for the first row is $S_1=(\mathrm{ADD}_3^{(1,3)}\mathrm{ADD}_3^{(1,5)}\mathrm{ADD}_3^{(1,6)}\mathrm{ADD}_3^{(1,7)}\mathrm{ADD}_3^{(1,8)})(M_2^{(5)}M_2^{(7)})$. The stabilizer matrix on application of operator S_1 becomes

$$\mathrm{H}_{\mathrm{CSS}}=\left[\begin{array}{cccccccc|cccccccc}1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\0&1&0&2&2&2&1&1&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&2&0&2&2&2&2\\0&0&0&0&0&0&0&0&0&1&0&2&2&2&1&1\end{array}\right]. \tag{B1}$$

Similarly, $S_2=\mathrm{ADD}_3^{(1,4)}\mathrm{ADD}_3^{(1,5)}\mathrm{ADD}_3^{(1,6)}\mathrm{ADD}_3^{(1,7)}\mathrm{ADD}_3^{(1,8)}(M_2^{(7)}M_2^{(8)})$ will transform $\mathrm{H}_{\mathrm{CSS}}$ as

$$\mathrm{H}_{\mathrm{CSS}}=\left[\begin{array}{cccccccc|cccccccc}1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0&2&0&2&2&1&1\\0&0&0&0&0&0&0&0&0&0&0&2&2&2&2&2\end{array}\right].$$

The operator $U_1=S_2S_1=\mathrm{ADD}_3^{(2,\{4,5,6,7,8\})}(M_2^{(7)}M_2^{(8)})\,\mathrm{ADD}_3^{(1,\{3,5,6,7,8\})}(M_2^{(5)}M_2^{(7)})$ is obtained using commutativity of MUL and ADD gates. On executing step 3 of Algorithm 1, we get $X=[\begin{smallmatrix}1&1&2&2\\1&1&1&1\end{smallmatrix}]$. On execution of step 4 of Algorithm 1, we can obtain the operator U_2 using the matrix X as $U_2=\mathrm{ADD}_3^{(8,4)}M_2^{(3)}\mathrm{ADD}_3^{(8,3)}M_2^{(3)}\mathrm{ADD}_3^{(7,4)}M_2^{(3)}\mathrm{ADD}_3^{(7,3)}M_2^{(3)}\,\mathrm{ADD}_3^{(6,4)}\mathrm{ADD}_3^{(6,3)}\mathrm{ADD}_3^{(5,4)}\mathrm{ADD}_3^{(5,3)}$, which can be shortened using commutativity of ADD and MUL gates

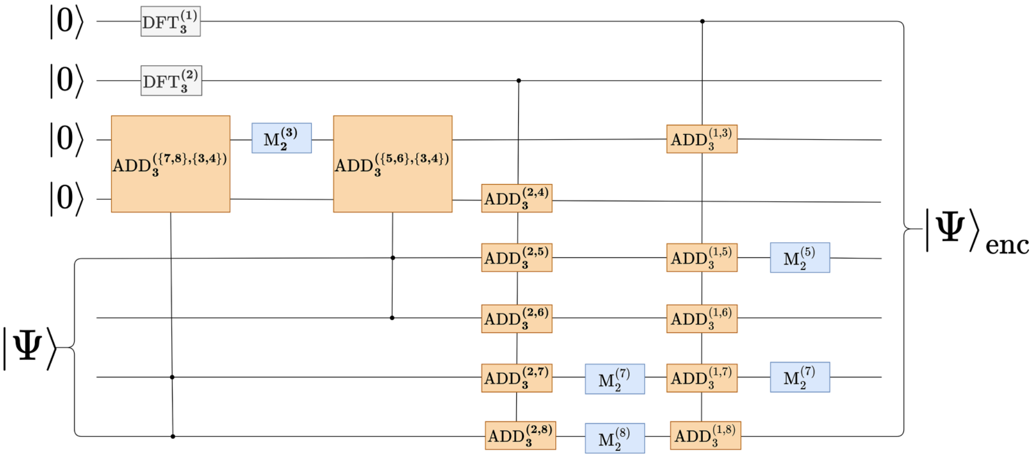


FIG. 11. Encoding circuit of a $[[8, 4, 3]]_3$ quantum code whose stabilizer generators are given in Eq. (B1). The state $|\Psi\rangle$ is encoded using a sequence of DFT, MUL, and ADD gates to obtain an encoded codeword $|\Psi\rangle_{\mathrm{enc}}$.

as $U_2 = M_2^{(3)}\mathrm{ADD}_3^{(\{7,8\},\{3,4\})}M_2^{(3)}\mathrm{ADD}_3^{(\{5,6\},\{3,4\})}$. The last $M_2^{(3)}$ gate in U_2 is redundant as it affects only the ancilla qutrit not the logical qutrit, so $U_2 = \mathrm{ADD}_3^{(\{7,8\},\{3,4\})}M_2^{(3)}\mathrm{ADD}_3^{(\{5,6\},\{3,4\})}$. The change in the stabilizer matrix is reflected as

$$\mathrm{H}_{\mathrm{CSS}} = \left[\begin{array}{cccccccc|cccccccc} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&2&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&2&0&0&0&0 \end{array}\right].$$

Finally, on executing step 4, the application of $\mathrm{DFT}_3^{\{1,2\}}$ gate on the first two qutrits will make $\mathrm{H}^{(x)}$ matrix zero, and the complete stabilizer matrix becomes

$$\mathrm{H}_{\mathrm{CSS}} = \left[\begin{array}{cccccccc|cccccccc} 0&0&0&0&0&0&0&0&2&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&2&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&2&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&2&0&0&0&0 \end{array}\right]. \tag{B2}$$

The complete encoding circuit is given in Fig. 11.

2. Example of hypergraph product code encoding

Now, we will provide an example where Algorithm 2 can be used to encode a hypergraph product code.

For simplicity, we consider a nondual containing $[3, 1]_2$ repetition code such that the parity check matrix is given by $\mathrm{H} = [\mathrm{I}_2|\mathrm{A}_1] = [\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\,|\,\begin{smallmatrix}1\\1\end{smallmatrix}]$. This is already in the row echelon form; hence, step 1 of Algorithm 2 is executed. An equivalent quantum code can be created using the hypergraph construction such that X and Z stabilizer matrices are given using Eqs. (84) and (85):

$$\mathrm{H}^{(x)} = [\mathrm{I}_2 \otimes \mathrm{I}_3|\mathrm{A}_1 \otimes \mathrm{I}_3|\mathrm{I}_2 \otimes \mathrm{H}^{\mathrm{T}}],$$

$$\mathrm{H}^{(x)} = \left[\begin{array}{cccccc|ccc|cccc} 1&0&0&0&0&0&1&0&0&1&0&0&0\\ 0&1&0&0&0&0&0&1&0&0&1&0&0\\ 0&0&1&0&0&0&0&0&1&1&1&0&0\\ 0&0&0&1&0&0&1&0&0&0&0&1&0\\ 0&0&0&0&1&0&0&1&0&0&0&0&1\\ 0&0&0&0&0&1&0&0&1&0&0&1&1 \end{array}\right], \tag{B3}$$

$$\mathrm{H}^{(z)} = \left[\mathrm{I}_3 \otimes \mathrm{H} \,\middle|\, \begin{array}{c} \mathrm{I}_2 \otimes \mathrm{I}_2 \\ \hline \mathrm{A}_1^{\mathrm{T}} \otimes \mathrm{I}_2 \end{array}\right],$$

$$\mathrm{H}^{(z)} = \left[\begin{array}{ccccccccc|cccc} 1&0&1&0&0&0&0&0&0&1&0&0&0\\ 0&1&1&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&1&0&1&0&0&0&0&0&1&0\\ 0&0&0&0&1&1&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0&1&1&0&1&0\\ 0&0&0&0&0&0&0&1&1&0&1&0&1 \end{array}\right]. \tag{B4}$$

On executing step 2 of Algorithm 2, we get

$$\begin{aligned} W_1 = {} & \mathrm{CNOT}^{\{6,\{9,12,13\}\}}\,\mathrm{CNOT}^{\{5,\{8,13\}\}}\,\mathrm{CNOT}^{\{4,\{7,12\}\}} \\ & \times \mathrm{CNOT}^{\{3,\{9,10,11\}\}}\,\mathrm{CNOT}^{\{2,\{8,11\}\}}\,\mathrm{CNOT}^{\{1,\{7,10\}\}}. \end{aligned} \tag{B5}$$

Using the operator W_1, the stabilizer matrices $\mathrm{H}^{(x)}$ and $\mathrm{H}^{(z)}$ will be transformed to

$$\mathrm{H}^{(x)} = \left[\begin{array}{cccccc|ccc|cccc} 1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0&0&0 \end{array}\right], \tag{B6}$$

$$\mathrm{H}^{(z)} = \left[\begin{array}{cccccc|ccc|cccc} 0&0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0&1&1&0&1&0\\ 0&0&0&0&0&0&0&1&1&0&1&0&1 \end{array}\right]. \tag{B7}$$

Now, step 3 of Algorithm 2 will provide us an operator $W_2 = \mathrm{CNOT}^{\{9,\{7,8\}\}}$, which will make the ninth column of $\mathrm{H}^{(z)}$ zero. We can also choose $W_2 = \mathrm{CNOT}^{\{7,\{8,9\}\}}$ that will make the seventh column zero of $\mathrm{H}^{(z)}$ matrix in Eq. (B7).

Finally, on executing step 4 of Algorithm 2, we will get an operator $U = \mathrm{H}^{\otimes 6}W_2W_1$ that transforms all the X stabilizer generators to identity and contains only Z stabilizer generators

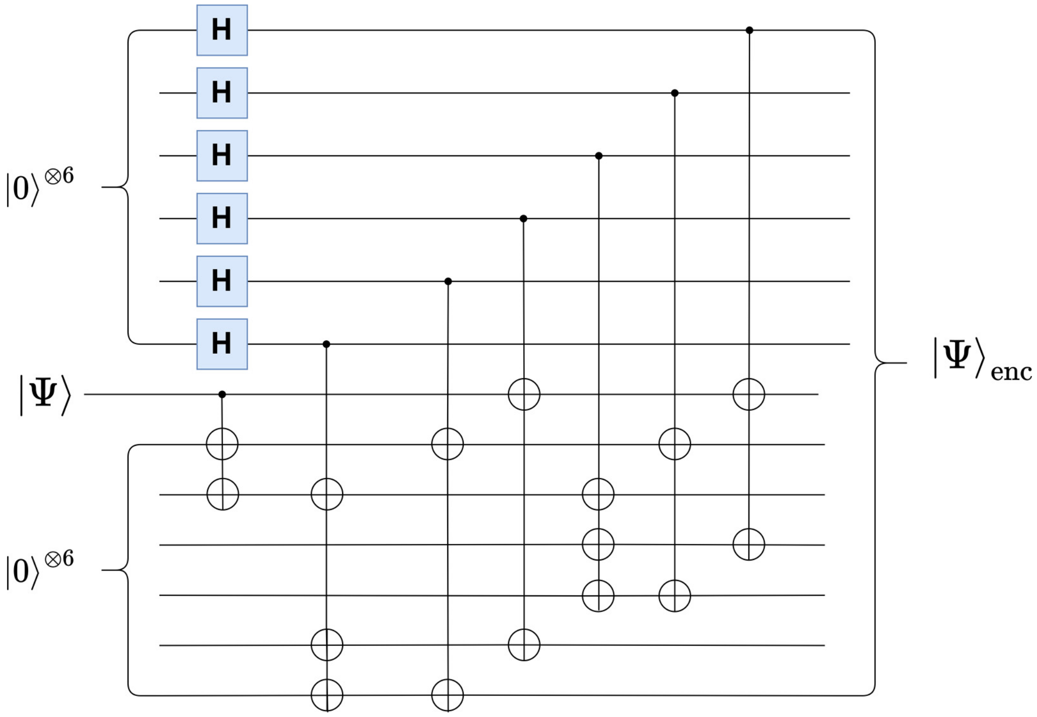


FIG. 12. The state $|\Psi\rangle$ is a single-qubit state encoded using an encoding circuit of the $[[13, 1, 3]]_2$ hypergraph product code containing only CNOT and Hadamard gates designed using the Algorithm 2 to obtain a 13-qubit encoded state $|\Psi\rangle_{\text{enc}}$.

in the matrix form

$$\mathrm{H}^{(z_f)} = \left[\begin{array}{cccccc|c|cccccc} 1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0&1&1&0&1&0\\ 0&0&0&0&0&0&0&1&1&0&1&0&1 \end{array}\right]. \tag{B8}$$

Finally, the complete stabilizer matrix is of the form $\mathrm{H}_S = [0_{12\times 13}|\mathrm{H}^{(z_f)}]$, where $0_{12\times 13}$ is a zero matrix. We will get the encoding operator $W = U^{\perp} = W_1W_2\mathrm{H}^{\otimes 6}$, using Algorithm 2. The encoding circuit of the $[[13, 1, 3]]_2$ hypergraph code is given in Fig. 12.

APPENDIX C: DECODERS FOR THE PROVIDED CODES

We provide the decoding operators for $[[31, 1, 16]]_2$ and $[[29, 1, 15]]_3$ codes using Algorithm 1, which are used for the distillation of the $|T\rangle$ and the $|S\rangle$ states, respectively. The decoding operators are helpful in the decoding step of the MSD procedure.

1. Decoding operator for the $[[31, 1, 16]]_2$ code

The parity check matrix of the [[31,1,16]] BCH code is given by

$$\mathrm{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6\cdots\alpha^{30}\\ 1 & \alpha^3 & \alpha^6 & \alpha^9 & \alpha^{12} & \alpha^{15} & \alpha^{18}\cdots\alpha^{28}\\ 1 & \alpha^5 & \alpha^{10} & \alpha^{15} & \alpha^{20} & \alpha^{25} & \alpha^{30}\cdots\alpha^{26} \end{bmatrix}, \tag{C1}$$

where α is the primitive element with primitive polynomial $x^5 + x^2 + 1$ over the field $\mathbb{F}_2^{31}$. We choose $\alpha = [0\,1\,0\,0\,0]^{\mathrm{T}}$ to transform the matrix H into the binary form as shown in Fig. 13(a). Now, we apply the steps of Algorithm 1 to get the decoding operator, which transforms the CSS stabilizer matrix

$$\mathrm{H}_{\mathrm{CSS}} = \left[\begin{array}{c|c} \mathrm{H}^{(x)} & 0\\ \hline 0 & \mathrm{H}^{(z)} \end{array}\right], \tag{C2}$$

where 0 is a 15×31 dimension-zero matrix and $\mathrm{H}^{(x)} = \mathrm{H}^{(z)} = \mathrm{H}$, to the matrix $\tilde{\mathrm{H}} = [0_{30\times 31}|S_{30\times 15}\,\tilde{S}_{30\times 15}\,0_{30\times 1}]$, where S and $\tilde{S}$ are some general matrices containing elements from $\mathbb{F}_2$.

In the first step of Algorithm 1, we apply Gaussian elimination on the matrix H to bring the matrix into row echelon form, i.e., $\tilde{\mathrm{H}} = [\mathrm{I}_{15\times 15}|\mathrm{A}_{15\times 15}|\mathrm{B}_{15\times 1}]$ as shown in Fig. 13(b) and assign X and Z stabilizer matrices with $\tilde{\mathrm{H}}$, i.e., $\mathrm{H}^{(x)} = \mathrm{H}^{(z)} = \tilde{\mathrm{H}}$ such that $\mathrm{H}^{(x)} = [\mathrm{I}^{(x)}|\mathrm{A}^{(x)}|\mathrm{B}^{(x)}]$ and $\mathrm{H}^{(z)} = [\mathrm{I}^{(z)}|\mathrm{A}^{(z)}|\mathrm{B}^{(z)}]$.

Now, in the second step of Algorithm 1, we iterate over the rows of $\mathrm{H}^{(x)} = [\mathrm{I}^{(x)}_{15\times 15}|\mathrm{A}^{(x)}_{15\times 15}|\mathrm{B}^{(x)}_{15\times 1}]$ matrix to get operators to make submatrix $[\mathrm{A}^{(x)}|\mathrm{B}^{(x)}]$ zero. For the first row, we apply $U_1 = \mathrm{CNOT}^{(1,c_1)}$, where $c_1 = \{16, 17, 20, 25, 26, 27, 28\}$ are the nonzero entries of the

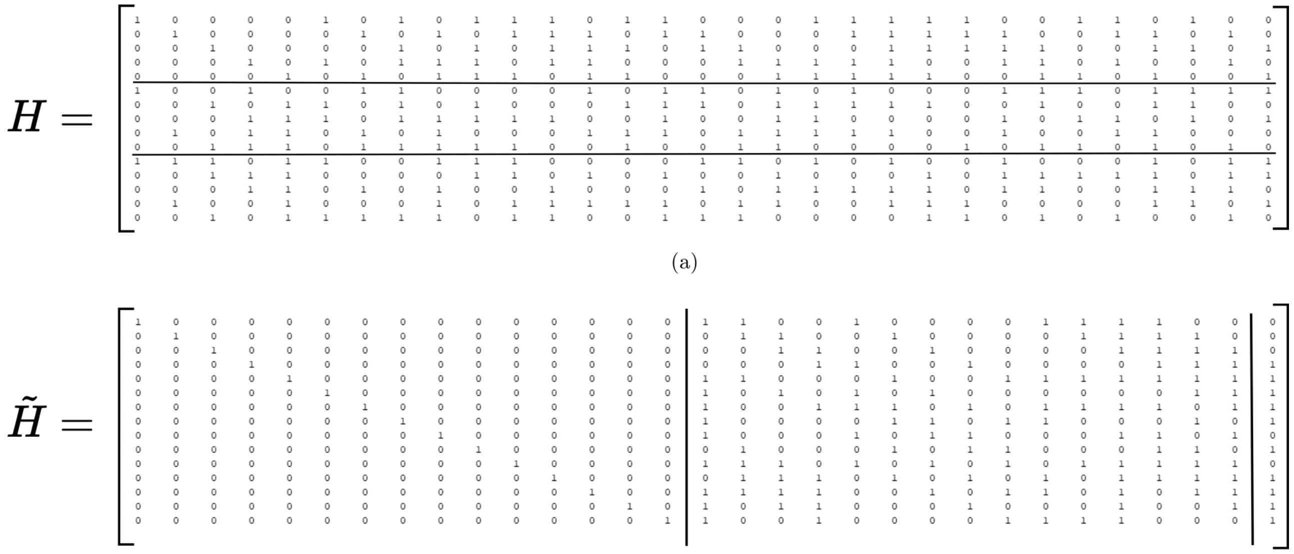


FIG. 13. (a) Shows the parity check matrix of the $[31, 16, 16]_2$ binary BCH self-dual code. (b) Shows the matrix after applying Gaussian elimination on the matrix in (a) required in step 1 of Algorithm 1.

row $[\mathrm{A}^{(x)}_{1,*}|\mathrm{B}^{(x)}_{1,*}]$ as shown in Fig. 13(b). Similarly, for the second row of $\mathrm{H}^{(x)}$, we apply $U_2 = \mathrm{CNOT}^{(1,c_2)}$ to make $[\mathrm{A}^{(x)}_{1,*}|\mathrm{B}^{(x)}_{1,*}]$ zero, where $c_2 = \{17, 18, 21, 26, 27, 28, 29\}$ are the nonzero entries of $[\mathrm{A}^{(x)}_{1,*}|\mathrm{B}^{(x)}_{1,*}]$ as shown in Fig. 13(b). On iterating over all the rows of $\mathrm{H}^{(x)}$, we get the operator $U = \prod_{i=1}^{15} \mathrm{CNOT}^{(i,c_i)}$ that makes the submatrix $[\mathrm{A}^{(x)}|\mathrm{B}^{(x)}]$ zero. The effect of the operator U will also make $\mathrm{I}^{(z)}$, the submatrix of $\mathrm{H}^{(z)}$ matrix zero, as proved earlier in Sec. V.

In the third step of Algorithm 1, we solve a linear equation $\mathrm{A}^{(z)}\underline{x} = \mathrm{B}^{(z)}$ and calculate the vector $\underline{x}$. The vector $\underline{x}$ helps to obtain an operator that will make the $\mathrm{B}^{(z)}$ submatrix of $\mathrm{H}^{(z)}$ zero. For the $[[31, 1, 16]]_2$ code, the vector $\underline{x} = [1\,0\,0\,0\,1\,1\,1\,1\,1\,0\,1\,0\,1\,1\,1]^{\mathrm{T}}$. The operator $U_2 = \mathrm{CNOT}^{(31,w)}$ will make $\mathrm{B}^{(z)}$ submatrix of $\mathrm{H}^{(z)}$ zero, where w are the target qubit positions calculated from the vector $\underline{x}$ by adding 15 to the indices i where $\underline{x}_i = 1$, for all $i \in \{1, 2, \dots, 15\}$, i.e, $w = \{16, 20, 21, 22, 23, 24, 26, 28, 29, 30\}$. Finally, we apply Hadamard gates on the first 15 qubits using an operator $T = \prod_{i=1}^{15} \mathrm{DFT}_2^{(i)}$, where $\mathrm{DFT}_2^{(i)}$ is a Hadamard gate applied on the ith qubit. The operator T will make the $\mathrm{I}^{(x)}$ submatrix of $\mathrm{H}^{(x)}$ zero. Finally, we get the decoding operator U for the $[[31, 1, 16]]_2$ code as $U = TU_2U_1$.

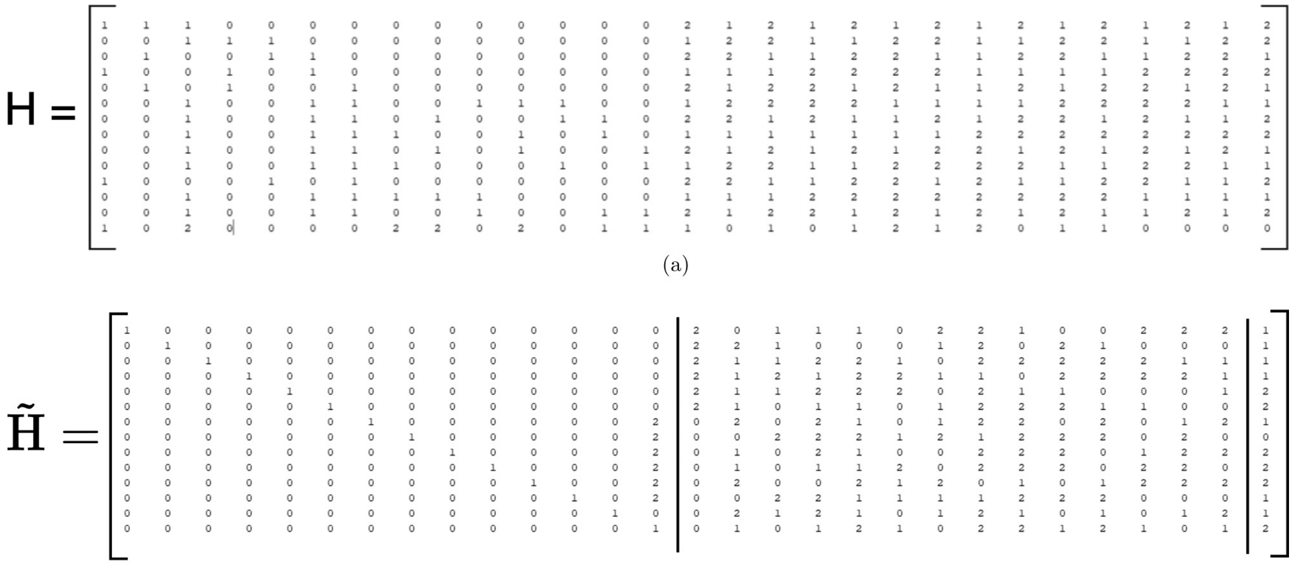


FIG. 14. (a) Shows the parity check matrix of the $[29, 15, 15]_3$ ternary self-dual code. (b) Shows the matrix after applying Gaussian elimination on matrix in (a) required in step 1 of Algorithm 1.

2. Decoding operator for the [[29, 1, 15]]$_3$ code

The parity check matrix H for the [[29, 1, 15]]$_3$ code is given in Fig. 14(a). On applying Gaussian elimination over the matrix H, we bring matrix H into row echelon form as $\tilde{\mathrm{H}} = [\mathrm{U}_{14\times14}|\mathrm{A}_{14\times14}|\mathrm{B}_{14\times1}]$ as shown in Fig. 14(b).

First assign $\tilde{\mathrm{H}}$ to $\mathrm{H}^{(x)} = [\mathrm{U}^{(x)}_{14\times14}|\mathrm{A}^{(x)}_{14\times14}|\mathrm{B}^{(x)}_{14\times1}]$ and $\mathrm{H}^{(z)} = [\mathrm{U}^{(z)}_{14\times14}|\mathrm{A}^{(z)}_{14\times14}|\mathrm{B}^{(z)}_{14\times1}]$ matrices and then iterate over the rows of $\mathrm{H}^{(x)}$. For the first row, we first apply multiplication gates (MUL) M_2 on the row $[\mathrm{A}^{(x)}_{1,*}|\mathrm{B}^{(x)}_{1,*}]$ where elements are 1 then we apply ADD$_3$ gates on all the nonzero positions of the updated $[\mathrm{A}^{(x)}_{1,*}|\mathrm{B}^{(x)}_{1,*}]$. The operator $o_1 = M_2^{(c_M^{(1)})}\mathrm{ADD}_3^{(1,c_A^{(1)})}$, $c_M^{(1)} = \{17, 18, 19, 23, 29\}$, and $c_A^{(1)} = \{15, 17, 18, 19, 21, 22, 23, 26, 27, 28, 29\}$. For the second row of $\mathrm{H}^{(x)}$, the operator $o_2 = M_2^{(c_M^{(2)})}\mathrm{ADD}_3^{(1,c_A^{(2)})}$, where $c_M^{(2)} = \{21, 25, 29\}$ and $c_A^{(2)} = \{15, 16, 17, 21, 22, 24, 25, 29\}$. Similarly, for any row i, o_i is calculated as $o_i = M_2^{(c_M^{(i)})}\mathrm{ADD}_3^{(1,c_A^{(i)})}$, where $c_M^{(i)}$ are the positions of $[\mathrm{A}^{(x)}_{i,*}|\mathrm{B}^{(x)}_{i,*}]$ where elements are 1 and $c_A^{(i)}$ are positions of all nonzero elements of $[\mathrm{A}^{(x)}_{i,*}|\mathrm{B}^{(x)}_{i,*}]$ which will make $[\mathrm{A}^{(x)}_{i}|\mathrm{B}^{(x)}_{i}]$ zero. As $\mathrm{U}^{(x)} \neq \mathrm{I}^{(x)}$, we need to add extra positions on c_A as $\tilde{c}_A^{(j)} = c_A \cup \{14\}$ for all $j \in \{7, 8, \dots, 12\}$. Finally, the operator $O_1 = \prod_{i=1}^{14} o_i$.

Next, we solve a linear equation $\mathrm{A}^{(z)}\underline{x} = \mathrm{B}^{(z)}$ to get $\underline{x} = [2\,0\,0\,1\,2\,2\,2\,2\,0\,0\,0\,1\,2\,2]$. The operator $O_2 = M_2^{(c_M)}\mathrm{ADD}^{(29,c_A)}$, where $c_M = \{i + 14\}$ such that $\underline{x}_i = 2$ and $c_A = \{j + 14\}$ such that $\underline{x}_j \neq 0$ for all $i, j \in S_{14} = \{1, 2, \dots, 14\}$, so $c_M = \{15, 19, 20, 21, 22, 27, 28\}$ and $c_A = c_M \cup \{18, 26\}$. Finally, DFT$_3$ gates are applied on the first 14 qutrits as follows: $T = \prod_{i=1}^{14} \mathrm{DFT}_3^{(i)}$ to make $U^{(x)}$ matrix zero. The complete decoding operator for the [[29, 1, 15]]$_3$ code is given by $O = TO_2O_1$ and the encoding operator $E = O^\dagger$.

[1] D. Gottesman, Stabilizer codes and quantum error correction, Ph.D. thesis, CalTech, 1997.

[2] P. J. Nadkarni and S. S. Garani, Non-binary entanglement-assisted stabilizer codes, Quantum Inf. Proc. **20**, 256 (2021).

[3] D. A. Lidar and T. A. Brun, *Quantum Error Correction* (Cambridge University Press, New York, 2013).

[4] H. K. Lau, R. Pooser, G. Siopsis, and C. Weedbrook, Quantum machine learning over infinite dimensions, Phys. Rev. Lett. **118**, 080501 (2017).

[5] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nat. Phys. **15**, 1273 (2019).

[6] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, Phys. Rev. A **71**, 022316 (2005).

[7] S. Prakash, Magic state distillation with the ternary Golay code, Proc. R. Soc. A **476**, 1471 (2020).

[8] D. Gottesman, Fault-tolerant quantum computation with higher-dimensional systems, *Quantum Computing and Quantum Communications*, Lecture Notes in Computer Science, Vol. 1509 (Springer-Verlag, Berlin, Heidelberg, 1999), pp. 302–313.

[9] S. Bravyi and J. Haah, Magic state distillation with low overhead, Phys. Rev. A **86**, 052329 (2012).

[10] H. Anwar, E. T. Campbell, and D. E. Browne, Qutrit magic state distillation, New J. Phys. **14**, 063006 (2012).

[11] M. Grassl, M. Rötteler, and T. Beth, Efficient quantum circuits for non-qubit quantum error-correcting codes, Int. J. Found. Comput. Sci. **14**, 757 (2003).

[12] P. J. Nadkarni and S. S. Garani, Encoding of nonbinary entanglement-unassisted and assisted stabilizer codes, IEEE Trans. Quantum Eng. **2**, 1 (2021).

[13] J.-P. Tillich and G. Zemor, Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength, IEEE Trans. Inf. Theory **60**, 1193 (2014).

[14] A. Jain and S. Prakash, Qutrit and ququint magic states, Phys. Rev. A **102**, 042409 (2020).

[15] S. Dutta, S. Banerjee, and M. Rani, Qudit states in noisy quantum channels, Phys. Scr. **98**, 115113 (2023).

[16] N. Rengaswamy, R. Calderbank, M. Newman, and H. D. Pfister, On optimality of CSS codes for transversal T, IEEE J. Selected Areas Inf. Theory **1**, 499 (2020).

[17] M. Webster, A. Quintavalle, and S. Bartlett, Transversal diagonal logical operators for stabiliser codes, New J. Phys. **25**, 103018 (2023).

[18] S. A. Aly, A note on quantum Hamming bound, arXiv:0711.4603.

[19] V. Veitch, C. Ferrie, D. Gross, and J. Emerson, Negative quasi-probability as a resource for quantum computation, New J. Phys. **14**, 113011 (2012).

[20] M. Grassl and T. Beth, Quantum BCH codes, arXiv:quant-ph/9910060.

[21] B. Gong, C. Ding, and C. Li, The dual codes of several classes of BCH codes, IEEE Trans. Inf. Theory **68**, 953 (2022).

[22] L. Joiner and J. Komo, Decoding binary BCH codes, in *Proceedings IEEE Southeastcon '95. Visualize the Future* (IEEE, Piscataway, NJ, 1995), pp. 67–73.

[23] S. Prakash and A. Gupta, Contextual bound states for qudit magic state distillation, Phys. Rev. A **101**, 010303(R) (2020).

[24] M. Howard, J. Wallman, V. Veitch, and J. Emerson, Contextuality supplies the 'magic' for quantum computation, Nature (London) **510**, 351 (2014).

[25] P. Panteleev and G. Kalachev, Degenerate quantum ldpc codes with good finite length performance, Quantum **5**, 585 (2021).

[26] P. Panteleev and G. Kalachev, Asymptotically good quantum and locally testable classical LDPC codes, STOC, arXiv:2111.03654.

[27] T. Richardson and S. Kudekar, Design of low-density parity check codes for 5G new radio, IEEE Commun. Mag. **56**, 28 (2018).

[28] S. S. Garani and B. Vasić, Channels engineering in magnetic recording: From theory to practice, IEEE BITS, Inf. Theory Mag. 1 (2023).

[29] S. S. Garani, L. Dolecek, J. Barry, F. Sala, and B. Vasić, Signal processing and coding techniques for 2-D magnetic recording: An overview, Proc. IEEE **106**, 286 (2018).