REALIZING LOGICAL DIAGONAL GATES VIA TRANSVERSAL PHYSICAL Z-ROTATIONS IN CSS CODES*

K. Sai Mineesh Reddy and Navin Kashyap
Dept. Electrical Communication Engineering, Indian Institute of Science
{mineeshk,nkashyap}@iisc.ac.in

ABSTRACT. Calderbank-Shor-Steane (CSS) codes, constructed from nested classical codes $C_2 \subseteq C_1$, are typically optimized for good code parameters. However, practical quantum computing equally demands fault-tolerant logical gates. In this work, we characterize nested pairs (C_1, C_2) whose resulting CSS codes realize a target logical diagonal gate via transversal physical Z-rotations. In doing so, we recover a result of Camps-Moreno et al. that CSS codes can realize only logical single-qubit Z-rotations and multi-qubit controlled-Z rotations via transversal physical Z-rotations. Building on our characterization, we develop the "appending construction", that takes as input an $[[n', k']]$ CSS code $\mathcal{Q}'$ and a target logical Z-rotation (single-qubit or multi-controlled) U_L, and extends $\mathcal{Q}'$ by systematically appending n'' physical qubits to obtain an $[[n, k]]$ CSS code $\mathcal{Q}$ with $n = n' + n''$ and $k = k'$. The target logical gate U_L is realized in $\mathcal{Q}$ by applying a well-chosen physical transversal Z-rotation to the n'' appended physical qubits. Moreover, any logical gate realized via transversal physical Z-rotations in the input code $\mathcal{Q}'$ remains transversally realizable in the extended code $\mathcal{Q}$. The CSS code $\mathcal{Q}$ may incur a loss in minimum distance, but the loss can be controlled through the parameter choices made in the construction. By repeatedly applying the appending construction, we can extend any CSS code $\mathcal{Q}'$ to obtain a CSS code $\mathcal{Q}$ that supports fault-tolerant implementations of multiple desired logical Z-rotations. The cost to be paid for this is the increased physical qubit overhead as the number of target logical gates grows.

1. INTRODUCTION

A central challenge in scalable quantum computing is designing quantum error-correcting codes [1] that simultaneously possess high encoding rates, robust error correction, and diverse fault-tolerant logical gates. Formally, a quantum code $\mathcal{Q}$ with parameters $[[n, k, d]]_2$ encodes k logical qubits (logical dimension) into n physical qubits (physical blocklength) and corrects Pauli errors of weight up to $\lfloor \frac{d-1}{2} \rfloor$. A logical operator is a unitary applied to the n physical qubits that preserves the code space $\mathcal{Q}$, effectively executing a logical gate on the k encoded qubits.

Because physical gates are inherently noisy, logical operators must be implemented fault-tolerantly. A canonical strategy for achieving this is the use of transversal gates (see [2, Chapter 5]). When a quantum system is partitioned into code blocks with independent error-correcting capabilities, a unitary is called transversal if it never couples physical qubits within the same block.

*This work was supported in part by the MATRICS grant ANRF/ARGM/2025/000814/MTR awarded by the Anusandhan National Research Foundation (ANRF).

This constraint ensures that physical gate failures remain localized: qubit errors cannot propagate within a block and are reliably corrected by the underlying code block. In this work, we adopt the simplest and most practical notion of transversality, wherein logical operators are implemented via tensor products of single-qubit unitaries.

Thus, an ideal objective is to find quantum codes capable of realizing a universal logical gate set entirely via transversal physical unitaries, but the Eastin-Knill theorem [3] forbids this. Quantum gates are typically partitioned into gates belonging to the Clifford group—generated by the Hadamard (H), Phase (S), and controlled-Z (CZ) gates—and non-Clifford gates. Because the Clifford group along with any single non-Clifford gate constitutes a universal gate set (see Section 2), no quantum code can transversally implement both simultaneously.

A natural relaxation is to find codes that transversally realize logical Clifford gates [4, 5, 6]. For instance, puncturing an $[n, n/2, d]$ classical self-dual code yields an $[[n-1, 1, \geq d-1]]$ Calderbank-Shor-Steane (CSS) code transversally supporting logical H, S, and logical CZ across any two such code blocks [7]. Considering ℓ such code blocks produces an $[[(n-1)\ell, \ell, \geq d-1]]$ code that transversally realizes the full logical Clifford group, meaning that logical H and S on each logical qubit, and logical CZ between any pair of logical qubits. Instantiating this with asymptotically good classical self-dual codes [8, 9] for n blocks yields a CSS family realizing the full logical Clifford group wherein both the number of logical qubits and minimum distance scale as the square root of the number of physical qubits. However, by the Gottesman-Knill theorem [10], Clifford-only circuits are efficiently simulatable on classical computers. Consequently, realizing only the Clifford group is insufficient for quantum advantage.

Typically, a diagonal gate such as the T (or the controlled-controlled-Z, i.e., CCZ) gate is selected as the required non-Clifford gate. Consequently, the set $\{H, \mathrm{CZ}, T\}$ (as $S = T^2$) constitutes a universal gate set. In practice, logical Clifford gates are executed transversally within one code, while non-Clifford gates are implemented via protocols like magic state distillation [11, 12] or code switching [13]. However, a crucial observation is that standard universal gate sets consist entirely of diagonal gates supplemented by a single superposition-creating Hadamard gate. This structure features prominently in major quantum algorithms, including Shor's factoring [14] and Grover's search [15] algorithms. Therefore, our methodology focuses on constructing codes that transversally realize a diverse set of logical diagonal gates, relying on code switching solely for the logical Hadamard gate. Because practical algorithms demand a rich variety of diagonal gates, this framework offers a viable paradigm for fault-tolerant quantum computation.

1.1. **Literature Review.** As motivated above, characterizing quantum codes that transversally implement logical diagonal gates is crucial for fault-tolerant quantum computing. Prior works [16, 17] investigate logical diagonal gates realized transversally within stabilizer codes. For CSS codes specifically, constructions derived from classical divisible codes transversally realize logical diagonal gates [18, 19, 20, 21]. Additionally, several studies establish characterizations, constructions, and necessary conditions for specific non-Clifford gates, primarily the T

gate [22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37] and multi-controlled-Z gates such as the CCZ gate [38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49].

While the aforementioned literature primarily focuses on algebraic constructions, another important line of research investigates topological codes. For instance, 2-D doubled color codes transversally realize the logical T gate on a single logical qubit [50]. Recently, this doubling technique was utilized to construct a class of quantum color codes encoding a single logical qubit that transversally realize any fixed logical single-qubit Z-rotation on the encoded qubit [51]. Furthermore, higher-dimensional color codes also transversally support logical diagonal gates, specifically the T gate [52, 53], the controlled-S (CS) gate [54], and the CCZ gate [55, 56, 57, 58, 59].

In this work, we characterize CSS codes that realize a fixed target logical diagonal gate via transversal physical Z-rotations. Camps-Moreno et al. [60] independently derived a closely related characterization. However, our methodologies differ: given a specific transversal physical Z-rotation, their approach determines whether it acts as a logical operator and computes the resulting logical diagonal gate using the code's logical-X basis. In contrast, our target-driven approach *a priori* fixes both the target logical diagonal gate and the transversal physical Z-rotation to determine the exact conditions required of the CSS code.

Furthermore, we specifically focus on realizing *addressable* logical gates, which selectively target specific subsets of logical qubits. Addressability is crucial for practical quantum algorithms, which require localized gates rather than logical gates acting on all logical qubits. Consequently, transversally realizing addressable logical gates is a major focus in the recent literature [61, 62, 63, 64, 65, 66] and in our current work.

1.2. **Our Main Contributions.** We now highlight the primary contributions of our paper.

1.2.1. *Characterization of CSS Codes That Realize Logical Diagonal Gates Via Transversal Z-Rotations.*

Restricting physical gates to dyadic transversal Z-rotations, we characterize the CSS codes capable of realizing a target logical diagonal gate. As defined in Sections 2.4 and 3.2, an n-qubit dyadic transversal Z-rotation $U(p, w)$ is a diagonal gate characterized by a non-negative integer p and an integer vector $w \in \mathbb{Z}^n$. We provide, in Theorem 5, a precise characterization of when $U(p, w)$ realizes a given logical diagonal gate within a specific CSS code. The theorem can be loosely paraphrased as follows:

> Let $\mathcal{Q}$ be an $[[n, k]]$ CSS code defined by a pair of nested binary linear codes $C_2 \subseteq C_1$, and let $U_{\mathrm{L}} = \mathrm{diag}\left(\exp\{\iota \frac{\pi}{2^\ell} f(a)\} : a \in \mathbb{F}_2^k\right)$ be a target k-qubit logical diagonal gate specified by an integer-valued function $f : \mathbb{F}_2^k \to \mathbb{Z}$. The transversal physical Z-rotation $U(p, w)$ realizes U_{L} if and only if $p \geq \ell$ and certain modular equations hold on the support of the vector w. These modular equations involve p, the codewords of C_2 and the coset space C_1/C_2, and the parameters f, ℓ that specify U_{L}.

Furthermore, as established by Camps-Moreno et al. [60] (see also Remark 5), CSS codes can realize only logical single-qubit Z-rotations and multi-controlled-Z rotations via transversal physical Z-rotations. Consequently, whenever we restrict physical gates to transversal Z-rotations, we assume that target gates consist entirely of single-qubit and multi-controlled-Z rotations.

Finally, we provide a relaxed characterization of CSS codes that realize a target logical diagonal gate via transversal physical Z-rotations. This relaxation allows the logical gate to be realized via transversal Z-rotations up to the application of physical Z gates.

1.2.2. *A Framework to Construct CSS Codes Transversally Realizing A Set of Logical Diagonal Gates.* While our characterization overlaps with prior literature [60], our primary contribution lies in leveraging it to propose a systematic constructive methodology: the appending framework. As established in Theorem 5, which we have paraphrased above, realizing a target logical gate U_{L} depends only on satisfying specific modular equations on the support of the physical Z-rotation's integer vector w. This localized dependency directly motivates our framework.

Given an $[[n, k]]$ primary CSS code and a set of target logical diagonal gates, we systematically append extra coordinates (physical qubits) by attaching auxiliary *appending matrices* to the X-stabilizer check matrix and logical-X generator matrix of the primary code. This process yields an extended CSS code wherein each target logical gate is assigned a dedicated subset of the newly appended coordinates. Realizing a specific target gate in this extended code involves choosing a transversal Z-rotation that acts exclusively on the physical qubits associated with the target logical gate's dedicated subset of extra coordinates. Furthermore, all logical diagonal gates realized via transversal physical Z-rotations in the primary code remain transversally realizable in the extended code.

The modularity of the appending framework allows the primary code to be flexibly extended to transversally support any desired set of logical diagonal gates. Crucially, this extended code preserves the number of logical qubits k while maintaining a minimum distance comparable to that of the primary code. However, on the flip side, the total number of physical qubits grows with the number of implemented logical gates.

1.2.3. *Appending Matrix Constructions for Addressable Single-Qubit and Multi-Controlled-Z Rotations.* To construct extended CSS codes capable of realizing any target logical gate achievable via transversal Z-rotations, we adopt an inductive construction strategy:

We first establish the appending matrix construction for addressable single-qubit Z-rotations, which serves as the base case. Building upon this, we systematically derive the appending matrices for addressable m-controlled-Z rotations by recursively leveraging the matrices for lower-order $\widetilde{m}$-controlled-Z rotations and single-qubit Z-rotations, where $\widetilde{m} < m$.

1.2.4. *Families of CSS Codes Realizing Addressable Single-Qubit Z-Rotations.* Leveraging our appending matrix constructions, we derive explicit CSS code families that transversally realize addressable logical single-qubit Z-rotations in two distinct regimes.

First, we consider an arbitrary but fixed sequence of target address configurations. In this regime, we obtain: (a) for $\ell = 1$, an asymptotically good CSS family with parameters $[[n, \Theta(n), \Omega(n)]]$ that transversally realizes the S gate on any fixed subset of logical qubits; and, (b) for $\ell > 1$, a CSS family with parameters $[[n, \Theta(n), \Omega(n^{\eta})]]$ that transversally realizes the Z-rotation $R_Z\left(\frac{\pi}{2^\ell}\right)$ on any fixed subset of at most n^ϵ logical qubits, where $\epsilon \in (0, 1)$ is a constant, and $\eta = \min\left\{\frac{1-\epsilon}{\ell}, \frac{1}{\ell+1}\right\}$.

Second, we consider codes that support *any* address configuration. For a fixed ℓ, we construct the following CSS families: (a) for $\ell = 1$, a family with parameters $[[n, \Omega(n^{1/2}), \Omega(n^{1/2})]]$ that transversally realizes any addressable logical S gate; and, (b) for $\ell > 1$, a family capable of transversally realizing any addressable logical $R_Z\left(\frac{\pi}{2^\ell}\right)$ gate, achieving parameters $[[n, \Omega(n^{1/(1+\epsilon)}), \Omega(n^{\epsilon/((\ell+1)(1+\epsilon))})]]$ when $\epsilon \in (0, \ell + 1)$, and $[[n, \Omega(n^{1/(1+\epsilon)}), \Omega(n^{1/(1+\epsilon)})]]$ when $\epsilon \geq \ell + 1$.

1.3. **Outline.** The paper is organized as follows. Section 2 establishes the notation and preliminaries. Section 3 parametrizes the transversal Z-rotations that preserve a CSS code and characterizes CSS codes that realize a target logical diagonal gate via transversal physical Z-rotations. Building on this, Section 4 introduces our general appending framework. Section 5 provides appending-matrix constructions for addressable logical single-qubit Z-rotations. For clarity of exposition, we focus on appending-matrix constructions for 1-controlled-Z rotations in Section 6 of the main text, deferring constructions for arbitrary multi-controlled-Z rotations to Appendix A. Section 7 applies these techniques to construct families of CSS codes that transversally realize addressable logical single-qubit Z-rotations. The paper has several appendices which contain proofs deferred from the main text.

2. Notation and Preliminaries

To lay the groundwork for our analysis, this section formalizes the core mathematical notation and preliminary concepts used throughout this work.

2.1. **Pauli Operators and the Clifford Hierarchy.** We begin with the single-qubit Pauli operators:

$$I_2 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad Y := \iota XZ = \begin{bmatrix} 0 & -\iota \\ \iota & 0 \end{bmatrix},$$

where $\iota = \sqrt{-1}$. These operators are Hermitian, unitary, and involutory (i.e., $X^2 = Y^2 = Z^2 = I_2$).

Let $\mathbb{F}_2$ denote the binary field. For any positive integer n and binary vectors $a = (a(1), \ldots, a(n)), b = (b(1), \ldots, b(n)) \in \mathbb{F}_2^n$, the n-qubit Pauli operators are defined via the tensor product

$$E(a, b) := \left(\iota^{a(1)b(1)} X^{a(1)} Z^{b(1)}\right) \otimes \cdots \otimes \left(\iota^{a(n)b(n)} X^{a(n)} Z^{b(n)}\right).$$

The n-qubit Pauli group is defined as the set

$$\mathcal{P}_n := \{\iota^\kappa E(a, b) : a, b \in \mathbb{F}_2^n, \kappa \in \{0, 1, 2, 3\}\}.$$

Like their single-qubit counterparts, the operators $E(a,b)$ are Hermitian, unitary, and involutory.

Setting $N = 2^n$, let $\mathbb{U}_N$ represent the group of all $N \times N$ unitaries. The Clifford hierarchy is defined recursively, starting with the Pauli group as its first level: $\mathcal{C}^{(1)} := \mathcal{P}_n$. For $\ell \geq 2$, the ℓ-th level is defined as the set of unitaries that map the Pauli group into the preceding level under conjugation:

$$\mathcal{C}^{(\ell)} := \left\{ U \in \mathbb{U}_N : UPU^\dagger \in \mathcal{C}^{(\ell-1)} \; \forall \; P \in \mathcal{P}_n \right\}.$$

The second level, $\mathcal{C}^{(2)}$, constitutes the Clifford group. It is generated by the Hadamard (H), Phase (S), and Controlled-NOT (CNOT) gates:

$$H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad S := \begin{bmatrix} 1 & 0 \\ 0 & \iota \end{bmatrix}, \qquad \text{CNOT} := \begin{bmatrix} I_2 & 0 \\ 0 & X \end{bmatrix}.$$

The controlled-Z (CZ) gate is also a Clifford gate given by[1]

$$\text{CZ} := (I_2 \otimes H) \cdot \text{CNOT} \cdot (I_2 \otimes H) = \begin{bmatrix} I_2 & 0 \\ 0 & Z \end{bmatrix}.$$

As is well-established, the Clifford group supplemented with any unitary from level $\ell \geq 3$ of the Clifford hierarchy forms a universal gate set for quantum computation.

2.2. **Binary Linear Codes and Asymptotics.** We next review standard coding theory notation. The Hamming weight of a binary vector x is denoted by $w_H(x)$. For a binary linear code C, we denote its dimension by $\dim(C)$ and its minimum distance by $d_{\min}(C)$. A code C is called m-divisible if every codeword $x \in C$ has Hamming weight $w_H(x) = 0 \pmod m$.

The Schur product of two vectors $x = (x(1), \ldots, x(n))$ and $y = (y(1), \ldots, y(n))$ in $\mathbb{F}_2^n$ is defined as their component-wise multiplication:

$$x * y := (x(1)y(1), \ldots, x(n)y(n)).$$

This operation extends naturally to codes. The Schur product of two codes C_1 and C_2 is defined as the $\mathbb{F}_2$-linear span of the pairwise Schur products of their respective codewords:

$$C_1 * C_2 := \langle x * y : x \in C_1, y \in C_2 \rangle.$$

Let $\mathbf{1}\{\cdot\}$ denote the indicator function and $[n] := \{1, 2, \ldots, n\}$. We denote the standard basis of $\mathbb{F}_2^n$ by $\{e_i : i \in [n]\}$, where e_i has a 1 at the i-th coordinate and 0 elsewhere. The support of a vector $x \in \mathbb{F}_2^n$ specifies the coordinates of its non-zero entries:

$$\text{Supp}(x) := \{i \in [n] : x(i) \neq 0\}.$$

Using inclusion-exclusion and induction, it can be shown that the binary linear combination $\bigoplus_{i=1}^k a(i)x_i$ (where $a(i) \in \mathbb{F}_2$ and $x_i \in \mathbb{F}_2^n$) can be expanded as an arithmetic sum over $\mathbb{Z}$ as

[1] The set $\{H, S, \text{CZ}\}$ also generates the Clifford group.

follows:

$$\bigoplus_{i=1}^{k} a(i)x_i = \sum_{i=1}^{k}(-2)^{i-1} \sum_{1\le j_1<\cdots<j_i\le k} \left(\prod_{p=1}^{i} a(j_p)\right)(x_{j_1} * \cdots * x_{j_i}),$$

where $\bigoplus$ and $\sum$ denote modulo-2 and integer addition, respectively. By the linearity of the dot product $w \cdot x = \sum_{i=1}^{n} w_i x_i$, for any integer vector $w \in \mathbb{Z}^n$, we have

$$w \cdot \left(\bigoplus_{i=1}^{k} a(i)x_i\right) = \sum_{i=1}^{k}(-2)^{i-1} \sum_{1\le j_1<\cdots<j_i\le k} \left(\prod_{p=1}^{i} a(j_p)\right) w \cdot (x_{j_1} * \cdots * x_{j_i}).$$

Let $\mathbf{1}_n$ and $\mathbf{0}_n$ denote the all-ones and all-zeros vectors of length n, respectively. Setting $w = \mathbf{1}_n$ yields the Hamming weight

$$w_H\left(\bigoplus_{i=1}^{k} a(i)x_i\right) = \sum_{i=1}^{k}(-2)^{i-1} \sum_{1\le j_1<\cdots<j_i\le k} \left(\prod_{p=1}^{i} a(j_p)\right) w_H(x_{j_1} * \cdots * x_{j_i}).$$

Crucially, if C is a 2^m-divisible code, the preceding equation implies that for any $\ell \in [k]$ and codewords $x_1, \ldots, x_\ell \in C$, we have

$$w_H(x_1 * \cdots * x_\ell) = 0 \pmod{2^{m-\ell+1}}.$$

To evaluate the asymptotic scaling of code parameters, we adopt the Bachmann-Landau order notations $\mathcal{O}$, Ω, and Θ as defined in [67].

2.3. **CSS Construction and Encoding.** Consider a nested pair of binary linear codes $C_2 \subseteq C_1$ with parameters $[n, k_2]$ and $[n, k_1]$, respectively.

The stabilizer group $\mathcal{S}$, generated by the X-type operators $E(x, 0)$ for $x \in C_2$ and the Z-type operators $E(0, z)$ for $z \in C_1^\perp$, defines an $[[n, k]]$ quantum code, where $k := k_1 - k_2$. The resulting CSS code is denoted by $(C_1, C_2)_{\text{CSS}}$ or $\mathcal{Q}_\mathcal{S}$. Its minimum distance is given by $d_{\min}(\mathcal{Q}_\mathcal{S}) := \min\{d_X, d_Z\}$, where the X- and Z-distances are given by

$$d_X := \min_{x \in C_1 \setminus C_2} w_H(x) \quad \text{and} \quad d_Z := \min_{z \in C_2^\perp \setminus C_1^\perp} w_H(z).$$

Here, $C_1 \setminus C_2$ denotes the set difference, distinct from the coset space C_1/C_2. The projector $P_\mathcal{S}$ onto the code space $\mathcal{Q}_\mathcal{S}$ is defined as

$$P_\mathcal{S} := \frac{1}{|C_2|}\left(\sum_{x \in C_2} E(x, 0)\right)\frac{1}{|C_1^\perp|}\left(\sum_{z \in C_1^\perp} E(0, z)\right).$$

Since the Z-type operators preserve $|0^n\rangle$, applying $P_\mathcal{S}$ to this state yields the logical all-zero basis state, $\left|0^k\right\rangle$:

$$\left|0^k\right\rangle_L := \sqrt{|C_2|} P_\mathcal{S} |0^n\rangle$$

$$= \frac{1}{\sqrt{|C_2|}} \sum_{x \in C_2} |x\rangle .$$

To construct all the remaining logical basis states $|a\rangle$ for $a \in \mathbb{F}_2^k \setminus \{0^k\}$, we first define logical-X operators. Any non-zero coset representative $x \in C_1/C_2$ acts as a non-trivial logical-X operator $E(x, 0)$. Therefore, we fix an arbitrary basis $\{y_1, \ldots, y_k\}$ for the coset space C_1/C_2.[2] Employing the standard overline notation $\overline{U}_{\mathrm{L}}$ to represent a physical operator implementing a corresponding logical gate U_{L} on the code space, the logical-X operator acting as the Pauli-X gate on the i-th logical qubit is defined as

$$\overline{X}_i := E(y_i, 0).$$

For any $a = (a(1), \ldots, a(k)) \in \mathbb{F}_2^k$, the multi-qubit logical-X gate acting over the qubits within the support of a is obtained by the product

$$\overline{X}_1^{a(1)} \cdots \overline{X}_k^{a(k)} = E(y_a, 0),$$

where $y_a := \bigoplus_{i=1}^k a(i) y_i$ is the coset representative corresponding to the vector a. Applying $E(y_a, 0)$ to the logical all-zero state yields the target logical basis state $|a\rangle$:

$$\begin{aligned} |a\rangle_L &:= \overline{X}_1^{a(1)} \cdots \overline{X}_k^{a(k)} \left|0^k\right\rangle_L = E(y_a, 0) \left|0^k\right\rangle_L \\ &= \frac{1}{\sqrt{|C_2|}} \sum_{x \in C_2} |x \oplus y_a\rangle . \end{aligned}$$

2.4. **Diagonal Gates and Z-Rotations.** We first formalize the notation for diagonal unitaries.

Definition 1. An n-qubit unitary operator U is *diagonal* if it acts on the computational basis as

$$U |x\rangle = \exp\{\iota \varphi_x\} |x\rangle , \quad \forall\, x \in \mathbb{F}_2^n,$$

where $\varphi_x \in \mathbb{R}$. We compactly denote such an operator by

$$U = \mathrm{diag}\left(\exp\{\iota \varphi_x\} : x \in \mathbb{F}_2^n\right).$$

Remark 1. Without loss of generality, we set $\varphi_{0^n} = 0$, ensuring that $U |0^n\rangle = |0^n\rangle$. Any non-zero φ_{0^n} introduces a phase factor $\exp\{\iota \varphi_{0^n}\}$, which can be factored out as a global phase.

Because the transversal gates we consider are tensor products of single-qubit unitaries, we first define single-qubit diagonal gates, namely Z-rotations.

Definition 2. A single-qubit Z-rotation by an angle $\theta \in [0, 2\pi)$ is defined by the unitary operator

$$R_Z(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & \exp\{\iota\theta\} \end{bmatrix}.$$

[2]Throughout this paper, the choice of coset representatives is not unique; any such choice should be regarded as arbitrary but fixed.

We term this a *dyadic Z-rotation* if the phase $\exp\{\iota\theta\}$ is a 2^ℓ-th root of unity (i.e., $\theta = \frac{2\pi k}{2^\ell}$ for some integer k) for a non-negative integer ℓ.

Notably, for $\ell \geq 2$, the dyadic Z-rotation $R_Z\left(\frac{\pi}{2^\ell}\right)$ is a non-Clifford gate. We next formalize multi-controlled-Z rotations.

Definition 3. For any positive integer m and non-negative integer ℓ, the m-controlled-Z rotation $C^{(m)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ is an $(m+1)$-qubit unitary that applies a phase of $\exp\left\{\iota\frac{\pi}{2^\ell}\right\}$ to a computational basis state if and only if all m control qubits and the target qubit are in the $|1\rangle$ state.

Since the operator $C^{(m)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ applies its phase exclusively to the all-ones state $|1\rangle^{\otimes(m+1)}$, the target and control qubits are completely interchangeable, rendering the action of the unitary invariant under any permutation of its input qubits. To specify its action on a particular subset of an ordered k-qubit system ($k \geq m+1$), we introduce the indexed notation $C^{(m)}_{i_1,\ldots,i_{m+1}}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$, where $1 \leq i_1 < \cdots < i_{m+1} \leq k$. By convention, we designate the highest-indexed qubit, i_{m+1}, as the target, while the first m indices, $i_1, \ldots, i_m$, identify the control qubits.

2.5. **Addressable Z-Rotations.** We conclude this section by formalizing the notation for the addressability of single-qubit and multi-controlled-Z rotations.

Definition 4. For a given binary vector $A \in \mathbb{F}_2^k$ and a dyadic Z-rotation $R_Z\left(\frac{\pi}{2^\ell}\right)$, we define the unitary $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ via the following k-fold tensor product:

$$R_Z\left(\frac{\pi}{2^\ell}\right)_A := \bigotimes_{i=1}^{k} R_Z\left(\frac{\pi}{2^\ell}A(i)\right) = \mathrm{diag}\left(\exp\left\{\iota\frac{\pi}{2^\ell}w_H(a*A) : a \in \mathbb{F}_2^k\right\}\right).$$

We refer to A as the *address vector* and $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ as the corresponding *addressed gate*.

Definition 5. Given an order-$(m+1)$ binary tensor $A \in \mathbb{F}_2^{k\times\cdots\times k}$ and an m-controlled-Z rotation $C^{(m)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$, we define the gate $C^{(m)}_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ as the following product over all ordered $(m+1)$-qubit subsets:

$$C^{(m)}_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right) := \prod_{1\leq i_1<\cdots<i_{m+1}\leq k} C^{(m)}_{i_1,\ldots,i_{m+1}}\left(R_Z\left(\frac{\pi}{2^\ell}A\left(i_1,\ldots,i_{m+1}\right)\right)\right).$$

The tensor A is referred to as the *address tensor*, and the unitary $C^{(m)}_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ is the corresponding *addressed gate*. Because the product is exclusively over $(m+1)$-tuples $(i_1,\ldots,i_{m+1})$ that are strictly increasing, the gate's action depends only on the tensor entries $A(i_1,\ldots,i_{m+1})$ for which $1 \leq i_1 < \cdots < i_{m+1} \leq k$. To eliminate ambiguity and to ensure that A is uniquely identified for an m-controlled-Z rotation, we enforce the convention that $A(i_1,\ldots,i_{m+1}) = 0$ unless we have $1 \leq i_1 < \cdots < i_{m+1} \leq k$.

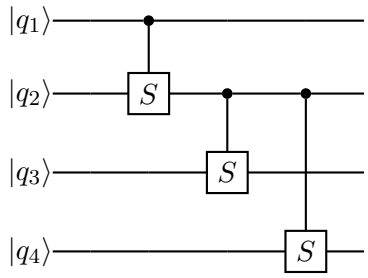
Example 1. To illustrate this framework, consider a 4-qubit register where the two-qubit interactions are controlled-S gates:

$$\mathsf{CS} := C^{(1)}\left(R_Z\left(\frac{\pi}{2}\right)\right).$$

Suppose the target addressability configuration is specified by the matrix $A \in \mathbb{F}_2^{4\times 4}$, whose only non-zero entries are $A(1,2) = A(2,3) = A(2,4) = 1$. The resulting addressed unitary, $\mathsf{CS}_A = C_A^{(1)}\left(R_Z\left(\frac{\pi}{2}\right)\right)$, evaluates to

$$\mathsf{CS}_A = \mathsf{CS}_{1,2} \cdot \mathsf{CS}_{2,3} \cdot \mathsf{CS}_{2,4},$$

where $\mathsf{CS}_{i,j} = C_{i,j}^{(1)}\left(R_Z\left(\frac{\pi}{2}\right)\right)$. The equivalent quantum circuit for this addressed operation is depicted below:



3. Characterizations of CSS Codes That Realize Logical Diagonal Gates Fault-Tolerantly

In this section, we characterize CSS codes that support the transversal implementation of a target logical diagonal gate.

3.1. Physical Diagonal Gates Realizing Logical Diagonal Gates.

The following theorem establishes the structural conditions under which a CSS code realizes a target logical diagonal gate through a fixed physical diagonal gate. Our proof leverages techniques introduced in [23] and independently in [60].

Theorem 1. *Let $(C_1, C_2)_{\mathsf{CSS}}$ be an $[[n,k]]$ CSS code. An n-qubit physical diagonal gate*

$$U_{\mathrm{P}} = \mathsf{diag}\left(\exp\{\iota\varphi_x\} : x \in \mathbb{F}_2^n\right)$$

realizes the k-qubit logical diagonal gate

$$U_{\mathrm{L}} = \mathsf{diag}\left(\exp\{\iota\lambda_a\} : a \in \mathbb{F}_2^k\right)$$

if and only if for all $x \in C_2$ and $a \in \mathbb{F}_2^k$, the following holds:

$$\exp\{\iota\varphi_{x\oplus y_a}\} = \exp\{\iota\lambda_a\}, \tag{1}$$

where $y_a \in C_1/C_2$ is the logical-X coset representative corresponding to the logical state $|a\rangle$.

Proof. The proof of sufficiency proceeds by direct substitution. For any $a \in \mathbb{F}_2^k$, we have

$$U_{\mathrm{P}} \left|a\right\rangle_L = \frac{1}{\sqrt{|C_2|}} \left(\sum_{x \in C_2} \exp\{\iota \varphi_{x \oplus y_a}\} \left|x \oplus y_a\right\rangle \right).$$

Applying the condition in (1), this simplifies to

$$U_{\mathrm{P}} \left|a\right\rangle_L \;=\; \frac{1}{\sqrt{|C_2|}} \left(\sum_{x \in C_2} \exp\{\iota \lambda_a\} \left|x \oplus y_a\right\rangle \right) \;=\; \exp\{\iota \lambda_a\} \left|a\right\rangle_L \;=\; \overline{U}_{\mathrm{L}} \left|a\right\rangle_L .$$

We now establish necessity. Suppose that the physical gate U_{P} realizes the target logical gate U_{L}. Then, for any $a \in \mathbb{F}_2^k$, we must have

$$U_{\mathrm{P}} \left|a\right\rangle_L = \gamma_{\mathrm{P}} \overline{U}_{\mathrm{L}} \left|a\right\rangle_L , \tag{2}$$

where γ_{P} denotes global phase. The left-hand side of (2) expands as

$$U_{\mathrm{P}} \left|a\right\rangle_L = \frac{1}{\sqrt{|C_2|}} \left(\sum_{x \in C_2} \exp\{\iota \varphi_{x \oplus y_a}\} \left|x \oplus y_a\right\rangle \right), \tag{3}$$

while the right-hand side evaluates to

$$\begin{aligned} \gamma_{\mathrm{P}} \overline{U}_{\mathrm{L}} \left|a\right\rangle_L &= \gamma_{\mathrm{P}} \exp\{\iota \lambda_a\} \left|a\right\rangle_L \\ &= \gamma_{\mathrm{P}} \exp\{\iota \lambda_a\} \frac{1}{\sqrt{|C_2|}} \left(\sum_{x \in C_2} \left|x \oplus y_a\right\rangle \right). \end{aligned} \tag{4}$$

Equating (3) and (4) and canceling the normalization factor $1/\sqrt{|C_2|}$, we find that for all $a \in \mathbb{F}_2^k$,

$$\sum_{x \in C_2} \exp\{\iota \varphi_{x \oplus y_a}\} \left|x \oplus y_a\right\rangle = \gamma_{\mathrm{P}} \exp\{\iota \lambda_a\} \left(\sum_{x \in C_2} \left|x \oplus y_a\right\rangle \right).$$

Because the states $\{|x \oplus y_a\rangle : x \in C_2\}$ form an orthonormal set in $\left(\mathbb{C}^2\right)^{\otimes n}$, equating the coefficients of the corresponding basis states implies that for all $x \in C_2$,

$$\exp\{\iota \varphi_{x \oplus y_a}\} = \gamma_{\mathrm{P}} \exp\{\iota \lambda_a\}.$$

In particular, setting $x = 0^n$ and $a = 0^k$ yields $\gamma_{\mathrm{P}} = 1$ (see Remark 1), completing the proof. □

3.2. **Physical Transversal Z-Rotations That Preserve a CSS Code.** For the remainder of this paper, we focus exclusively on CSS codes that realize logical diagonal gates transversally.

The following theorem establishes that transversal Z-rotations preserving a CSS code space are necessarily dyadic.

Theorem 2 ([17], Theorem 1)**.** *If $U_{\mathrm{P}} = \bigotimes_{i=1}^n R_Z(\theta_i)$ is a logical operator on a CSS code space, then each phase $\exp\{\iota \theta_i\}$ must be a 2^{l_i}-th root of unity for some non-negative integer l_i.*

Remark 2. As a consequence of the preceding theorem, we hereafter restrict our attention to dyadic transversal Z-rotations of the form

$$U_{\mathrm{P}} = \bigotimes_{i=1}^{n} R_Z\left(\frac{\pi q_i}{2^{p_i}}\right),$$

where each p_i is a non-negative integer and each q_i is an odd integer.

The following lemma demonstrates that every transversal Z-rotation can be parameterized by a non-negative integer and an integer vector.

Lemma 3. *Any n-qubit dyadic transversal Z-rotation is characterized by a non-negative integer p and an integer vector $w \in \mathbb{Z}^n$ containing at least one odd component, such that*

$$U = \mathrm{diag}\left(\exp\left\{\iota\frac{\pi}{2^p}(w \cdot x)\right\} : x \in \mathbb{F}_2^n\right). \tag{5}$$

Moreover, the integer p is uniquely determined, and the vector w is unique modulo 2^{p+1}.

The proof of this lemma is provided in Appendix C. Under this parameterization, the diagonal unitary U resides precisely in the $(p+1)$-th level of the Clifford hierarchy, $\mathcal{C}^{(p+1)}$.

Motivated by the characterization established in Lemma 3, we introduce the following formal parameterization for transversal Z-rotations.

Definition 6. Any n-qubit dyadic transversal Z-rotation is uniquely identified by a non-negative integer p and an integer vector $w \in \mathbb{Z}^n$ containing at least one odd component. We denote this operator by the tuple (p, w), explicitly defined as

$$U(p, w) := \mathrm{diag}\left(\exp\left\{\iota\frac{\pi}{2^p}(w \cdot x)\right\} : x \in \mathbb{F}_2^n\right).$$

3.3. **CSS Codes Realizing Logical Diagonal Gates Fault-Tolerantly.** In this subsection, we characterize CSS codes capable of realizing a targeted logical diagonal gate via transversal physical Z-rotations. Previously, Camps-Moreno et al. [60] established similar conditions under which transversal physical Z-rotations preserve the code space, determined their resulting logical actions, and identified the operator groups that induce non-trivial logical gates versus those that induce the logical identity. In contrast, the following lemma simultaneously fixes both the transversal physical operations and the target logical gate.

Lemma 4. *Let $(C_1, C_2)_{\mathrm{CSS}}$ be an $[[n, k]]$ CSS code. An n-qubit transversal physical Z-rotation $U(p, w)$ realizes the k-qubit logical diagonal gate*

$$U_{\mathrm{L}} = \mathrm{diag}\left(\exp\{\iota\lambda_a\} : a \in \mathbb{F}_2^k\right)$$

if and only if for all $x \in C_2$ and $a \in \mathbb{F}_2^k$, the following condition holds:

$$\exp\left\{\iota\frac{\pi}{2^p}(w \cdot (x \oplus y_a))\right\} = \exp\{\iota\lambda_a\},$$

where $y_a \in C_1/C_2$ is the logical-X representative corresponding to the logical state $|a\rangle$.

Proof. The result follows directly by substituting the parameterization of $U(p, w)$ from Definition 6 into the condition established in Theorem 1. □

The preceding characterization implies that each phase exponent λ_a must be of the form $\frac{\pi m_a}{2^p}$ for some integer m_a. The following theorem refines this characterization by explicitly incorporating the discrete phase constraint.

Theorem 5. *Let $(C_1, C_2)_{\text{CSS}}$ be an $[[n, k]]$ CSS code, and let $f : \mathbb{F}_2^k \to \mathbb{Z}$ be an integer-valued function such that $f(0^k) = 0 \pmod{2^{\ell+1}}$ and $f(a_0)$ is odd for some $a_0 \in \mathbb{F}_2^k$. An n-qubit transversal physical Z-rotation $U(p, w)$ realizes the k-qubit logical diagonal gate*

$$U_{\text{L}} = \text{diag}\left(\exp\left\{\iota \frac{\pi}{2^\ell} f(a)\right\} : a \in \mathbb{F}_2^k\right)$$

if and only if $p \geq \ell$ and for all $x \in C_2$ and $a \in \mathbb{F}_2^k$, the following modular equations hold:

$$w \cdot x = 0 \pmod{2^{p+1}}, \tag{6}$$

$$w \cdot y_a = 2^{p-\ell} f(a) \pmod{2^{p+1}}, \tag{7}$$

$$w \cdot (x * y_a) = 0 \pmod{2^p}, \tag{8}$$

where $y_a \in C_1/C_2$ is the logical-X representative corresponding to the logical state $|a\rangle$.

Proof. By Lemma 4, the physical gate $U(p, w)$ realizes the logical gate U_{L} if and only if for all $x \in C_2$ and $a \in \mathbb{F}_2^k$,

$$\exp\left\{\iota \frac{\pi}{2^p} (w \cdot (x \oplus y_a))\right\} = \exp\left\{\iota \frac{\pi}{2^\ell} f(a)\right\}. \tag{9}$$

In particular, when this expression is evaluated at $a = a_0$, the condition that $f(a_0)$ is odd necessitates that $p \geq \ell$. Consequently, condition (9) is equivalent to

$$w \cdot (x \oplus y_a) = 2^{p-\ell} f(a) \pmod{2^{p+1}}. \tag{10}$$

For the remainder of the proof, we first establish necessity. The left-hand side of (10) can be expanded as

$$w \cdot x + w \cdot y_a - 2\,(w \cdot (x * y_a)) = f(a) \pmod{2^{p+1}}. \tag{11}$$

Since $f(0^k) = 0 \pmod{2^{p+1}}$, setting $a = 0$ (which implies $y_a = 0$) in (11) yields (6):

$$w \cdot x = 0 \pmod{2^{p+1}}. \tag{12}$$

Similarly, setting $x = 0$ in (11) isolates (7):

$$w \cdot y_a = f(a) \pmod{2^{p+1}}. \tag{13}$$

Substituting (12) and (13) back into (11), we find that $-2\,(w \cdot (x * y_a)) = 0 \pmod{2^{p+1}}$, yielding (8). To prove sufficiency, we simply substitute the conditions (6)–(8) back into the left-hand side of (11), which immediately recovers the original modular equation (10). □

Remark 3. From the above characterization, it is immediate that within a CSS code, a physical rotation $U(p,w)$ realizes the logical gate U_{L} if and only if $U(p,-w)$ realizes $U_{\mathrm{L}}^{\dagger}$. This follows directly from the diagonal representation

$$U_{\mathrm{L}}^{\dagger} = \mathrm{diag}\left(\exp\Big\{-\iota\frac{\pi}{2^{\ell}}f(a)\Big\} : a \in \mathbb{F}_2^k\right).$$

Remark 4. The modular criteria in (6) and (8) merely guarantee that the physical rotation $U(p,w)$ acts as a logical operator. To see this, we apply the gate to an arbitrary logical basis state $|a\rangle$:

$$\begin{aligned} U(p,w)\,|a\rangle_L &= \frac{1}{\sqrt{|C_2|}}\sum_{x\in C_2}\exp\Big\{\iota\frac{\pi}{2^p}w\cdot(x\oplus y_a)\Big\}\,|x\oplus y_a\rangle \\ &= \frac{1}{\sqrt{|C_2|}}\sum_{x\in C_2}\exp\Big\{\iota\,\frac{\pi}{2^p}\,(w\cdot x - 2w\cdot(x*y_a) + w\cdot y_a)\Big\}\,|x\oplus y_a\rangle \\ &= \exp\Big\{\iota\,\frac{\pi}{2^p}\;(w\cdot y_a)\Big\}\,|a\rangle_L\,, \end{aligned}$$

the final equality being obtained by an application of (6) and (8). Thus, realizing a specific target logical gate U_{L} depends entirely on satisfying the modular equation (7) over the coset space.

The following remark provides a general formula for the logical diagonal gates realizable via transversal physical Z-rotations within a CSS code. From this, we observe that such logical gates are limited to single-qubit Z-rotations and multi-controlled-Z rotations. Both this formula and the observation were independently derived by Camps-Moreno et al. [60].

Remark 5. The function f in Theorem 5 cannot be chosen arbitrarily. This restriction arises because f expands as

$$2^{p-\ell}f(a) = w\cdot y_a = \sum_{i=1}^{k}(-2)^{i-1}\sum_{1\le j_1<\cdots<j_i\le k}\left(\prod_{q=1}^{i}a(j_q)\right)w\cdot(y_{j_1}*\cdots*y_{j_i}) \pmod{2^{p+1}}.$$

Consequently, the logical gate U_{L} decomposes into a product of single-qubit dyadic Z-rotations and multi-controlled dyadic Z-rotations, whose realization is governed entirely by the coset basis $\{y_1,\ldots,y_k\}$ and its t-fold Schur products (for $t\ge 2$):

$$\begin{aligned} U_{\mathrm{L}} = \left(\prod_{i=1}^{k}R_Z\left(\frac{\pi}{2^p}w\cdot y_i\right)_{e_i}\right)\cdot\left(\prod_{1\le j_1<j_2\le k}C_{j_1,j_2}\left(R_Z\left(-\frac{\pi}{2^{p-1}}w\cdot(y_{j_1}*y_{j_2})\right)\right)\right) \\ \cdots\; C_{1,\ldots,k}\left(R_Z\left((-1)^{k-1}\frac{\pi}{2^{p-k+1}}w\cdot(y_1*\cdots*y_k)\right)\right). \end{aligned}$$

For certain choices of the function $f:\mathbb{F}_2^k\to\mathbb{Z}$, to satisfy the conditions (6)–(8) in Theorem 5, a stronger condition than $p\ge\ell$ is needed. For example, take $k=3$ and consider the function $f(a)=1$ if $a=111$ and $f(a)=0$ otherwise. The logical gate defined by this f is the 2-controlled Z-rotation $C^{(2)}\left(R_Z(\frac{\pi}{2^{\ell}})\right)$. In particular, for $\ell=0$, this is the CCZ gate. Observe by Theorem 5 that

a transversal physical Z-rotation $U(p,w)$ realizes logical CCZ only if $p \geq 0$. However, for $p \leq 1$, transversal physical Z-rotations lie within the Clifford group, whereas CCZ is a non-Clifford gate in the third level of the Clifford hierarchy. So, one intuitively expects that $U(p,w)$ can realize CCZ only if $p \geq 2$. Indeed, Proposition 6 below shows that the inequality $p \geq \ell$ must be further strengthened for $U(p,w)$ to realize an addressable multi-controlled Z-rotation.

Let $(C_1, C_2)_{\text{CSS}}$ be an $[[n,k]]$ CSS code, and consider an addressable logical $(m-1)$-controlled-$R_Z\left(\frac{\pi}{2^\ell}\right)$ gate acting on the k-qubit code space, where $2 \leq m \leq k$. Parameterized by an order-m target address tensor $A \in \mathbb{F}_2^{k\times\cdots\times k}$, the logical gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ is given by

$$\begin{aligned} C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right) &= \prod_{1\leq i_1<\cdots<i_m\leq k} C_{i_1,\ldots,i_m}^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}A\left(i_1,\ldots,i_m\right)\right)\right) \\ &= \operatorname{diag}\left(\exp\left\{\iota\frac{\pi}{2^\ell}\left(\sum_{1\leq i_1<\cdots<i_m\leq k} A\left(i_1,\ldots,i_m\right)a\left(i_1\right)\cdots a\left(i_m\right)\right)\right\} : a\in\mathbb{F}_2^k\right), \end{aligned}$$

where at least one upper-triangular tensor entry $A(i_1,\ldots,i_m)$ is odd.

Proposition 6. *An n-qubit transversal physical Z-rotation $U(p,w)$ realizes the logical gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ on $(C_1,C_2)_{\text{CSS}}$ if and only if $p \geq \ell+m-1$ and for all $x \in C_2$ and $a \in \mathbb{F}_2^k$, the following modular condition holds:*

$$\begin{aligned} w\cdot x &= 0 \pmod{2^{p+1}} \\ w\cdot y_a &= 2^{p-\ell}\left(\sum_{1\leq i_1<\cdots<i_m\leq k} A\left(i_1,\ldots,i_m\right)a\left(i_1\right)\cdots a\left(i_m\right)\right) \pmod{2^{p+1}}, \\ w\cdot(x*y_a) &= 0 \pmod{2^p} \end{aligned}$$

where $y_a \in C_1/C_2$ is the logical-X representative corresponding to the logical state $|a\rangle$.

The proof of this proposition is deferred to Appendix D.

4. Appending Framework: Constructing CSS Codes That Realize Logical Diagonal Gates Fault-Tolerantly

In this section, we propose a systematic framework for extending a CSS code to realize an arbitrary number of target logical diagonal gates (Z-rotations or multi-controlled Z-rotations) via transversal physical Z-rotations. Specifically, given an $[[n',k']]$ primary CSS code and an arbitrary but fixed sequence of logical diagonal gates $U_{\text{L}}^{(1)},\ldots,U_{\text{L}}^{(M)}$ (for any $M\in\mathbb{N}$) as described in Theorem 5, our objective is to construct a "derived" CSS code that preserves the k' logical qubits while realizing these logical gates transversally. Furthermore, we require this derived code to maintain a minimum distance comparable to that of the primary code.

Consider a primary CSS code, denoted by $(C_1', C_2')_{\text{CSS}}$, with parameters $[[n', k', \geq d']]$, where the minimum X- and Z-distances satisfy

$$d'_X = \min_{x \in C_1' \setminus C_2'} w_H(x) \geq d',$$
$$d'_Z = \min_{z \in (C_2')^\perp \setminus (C_1')^\perp} w_H(z) \geq d_{\min}((C_2')^\perp) \geq d'.$$

Let $G_{C_2'}$ and $G_{C_1'/C_2'}$ denote the generator matrices for the code C_2' and the coset space C_1'/C_2', respectively. The generator matrix for C_1', denoted by $G_{C_1'}$, can then be expressed as the block matrix

$$G_{C_1'} = \begin{bmatrix} G_{C_1'/C_2'} \\ G_{C_2'} \end{bmatrix}.$$

For each $i \in \mathbb{N}$, suppose the diagonal representation of $U_{\text{L}}^{(i)}$ is given by

$$U_{\text{L}}^{(i)} = \text{diag}\left(\exp\left\{\iota \frac{\pi}{2^{\ell_i}} f_i(a)\right\} : a \in \mathbb{F}_2^{k'}\right),$$

where ℓ_i is a non-negative integer and $f_i : \mathbb{F}_2^{k'} \to \mathbb{Z}$ is a function as in Theorem 5.

The primary CSS code $(C_1', C_2')_{\text{CSS}}$ need not inherently realize any of the target gates $U_{\text{L}}^{(i)}$ transversally. To handle this, for each target logical gate $U_{\text{L}}^{(i)}$, we append dedicated matrices to the generator matrices $G_{C_2'}$ and $G_{C_1'/C_2'}$ of the primary CSS code to obtain augmented generator matrices G_{C_2} and G_{C_1/C_2} that define our derived CSS code $(C_1, C_2)_{\text{CSS}}$. Each appended matrix is structured to ensure that the modular equations (6)–(8) required to realize the specific logical gate $U_{\text{L}}^{(i)}$ in $(C_1, C_2)_{\text{CSS}}$ are satisfied for some transversal physical Z-rotation $U(p_i, w_i)$. Notably, any logical diagonal gate realized via transversal physical Z-rotations in the primary code remains transversally realizable within the derived code, since the corresponding physical Z-rotation applied to the primary code's physical qubits in the derived code implements the exact same logical gate.

To execute this plan, carefully chosen auxiliary matrices $G_1^{(1)} \in \mathbb{F}_2^{k' \times n_1}, \ldots, G_1^{(M)} \in \mathbb{F}_2^{k' \times n_M}$ are sequentially appended to $G_{C_1'/C_2'}$, where each $G_1^{(i)}$ is dedicated to realizing the logical gate $U_{\text{L}}^{(i)}$. (The issue of precisely how these matrices $G_1^{(i)}$ are chosen will be dealt with later.) Thus, we have

$$G_{C_1/C_2} = \begin{bmatrix} G_{C_1'/C_2'} & G_1^{(1)} & G_1^{(2)} & \cdots & G_1^{(M)} \end{bmatrix}.$$

Appending the matrix $G_{C_2'}$ with zeros will result in a trivial Z-distance ($d_Z = 1$) for the resulting CSS code. Instead, we augment $G_{C_2'}$ with another carefully chosen set of auxiliary matrices $G_2^{(1)}, \ldots, G_2^{(M)}$, each $G_2^{(i)}$ being a $k_2^{(i)} \times n_i$ matrix over $\mathbb{F}_2$. Note that $G_1^{(i)}$ and $G_2^{(i)}$ have the same

number of columns so that one can be stacked atop the other. Thus,

$$G_{C_2} = \begin{bmatrix} G_{C_2'} & 0 & 0 & \cdots & 0 \\ 0 & G_2^{(1)} & 0 & \cdots & 0 \\ 0 & 0 & G_2^{(2)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & G_2^{(M)} \end{bmatrix}$$

and

$$G_{C_1} = \begin{bmatrix} G_{C_1/C_2} \\ G_{C_2} \end{bmatrix} = \begin{bmatrix} G_{C_1'/C_2'} & G_1^{(1)} & G_1^{(2)} & \cdots & G_1^{(M)} \\ G_{C_2'} & 0 & 0 & \cdots & 0 \\ 0 & G_2^{(1)} & 0 & \cdots & 0 \\ 0 & 0 & G_2^{(2)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & G_2^{(M)} \end{bmatrix}.$$

The lemma below, proved in Appendix F, establishes the parameters of the derived code $(C_1, C_2)_{\text{CSS}}$.

Lemma 7. *$(C_1, C_2)_{\text{CSS}}$ is an $[[n' + \sum_{i=1}^{M} n_i, k', \min\{d_X, d_Z\}]]$ code, where $d_X \geq d'$ and*

$$d_Z \geq \min\{d_{\min}((C_2')^{\perp}), d_{\min}((C_2^{(1)})^{\perp}), \ldots, d_{\min}((C_2^{(M)})^{\perp})\},$$

with $C_2^{(i)}$ being the code generated by the rows of $G_2^{(i)}$ for each $i \in [M]$.

We now establish the precise conditions on the matrices $G_1^{(i)}$ and $G_2^{(i)}$ required to realize $U_{\text{L}}^{(i)}$ via a transversal Z-rotation $U(p_i, w_i)$ applied to the physical qubits of $(C_1, C_2)_{\text{CSS}}$. This rotation is applied exclusively to the physical qubits corresponding to the block-column of G_{C_1} that contains $G_1^{(i)}$ and $G_2^{(i)}$. Specifically, the vector w_i takes the form

$$w_i = \left(\mathbf{0}_{n'}, \mathbf{0}_{n_1}, \ldots, \mathbf{0}_{n_{i-1}}, \widetilde{w}_i, \mathbf{0}_{n_{i+1}}, \ldots, \mathbf{0}_{n_M}\right).$$

Let $y_1', \ldots, y_{k'}'$ and $y_1^{(i)}, \ldots, y_{k'}^{(i)}$ denote the row vectors of the matrices $G_{C_1'/C_2'}$ and $G_1^{(i)}$, respectively. The appending procedure merges these components so that the rows of G_{C_1/C_2} take the block form

$$y_j = \left(y_j', y_j^{(1)}, \ldots, y_j^{(M)}\right) \text{ for } j \in [k'].$$

Because C_2 is constructed as a direct sum, for any $x \in C_2$, there exist $x' \in C_2'$ and $x_i \in C_2^{(i)} :=$ rowspace$(G_2^{(i)})$ such that $x = (x', x_1, \ldots, x_M)$. Moreover, for any $a \in \mathbb{F}_2^{k'}$, the corresponding logical-X representative is given by $y_a = \left(y_a', y_a^{(1)}, \ldots, y_a^{(M)}\right)$, where

$$y_a' = \bigoplus_{j=1}^{k'} a(j) y_j' \quad \text{and} \quad y_a^{(i)} = \bigoplus_{j=1}^{k'} a(j) y_j^{(i)} \text{ for } i \in [M].$$

By Theorem 5 and the preceding decomposition, the physical gate $U(p_i, w_i)$ realizes the logical gate $U_{\mathrm{L}}^{(i)}$ if and only if for all $x_i \in C_2^{(i)}$ and $a \in \mathbb{F}_2^{k'}$, the following modular constraints hold:

$$\begin{aligned} \widetilde{w}_i \cdot x_i &= 0 \pmod{2^{p_i+1}} \\ \widetilde{w}_i \cdot y_a^{(i)} &= 2^{p_i-\ell_i} f_i(a) \pmod{2^{p_i+1}} \\ \widetilde{w}_i \cdot \left(x_i * y_a^{(i)}\right) &= 0 \pmod{2^{p_i}}. \end{aligned} \tag{14}$$

These conditions make it clear that realizing the logical gate $U_{\mathrm{L}}^{(i)}$ within the code $(C_1, C_2)_{\mathsf{CSS}}$ depends entirely on the design of the auxiliary matrices $G_1^{(i)}$ and $G_2^{(i)}$, as well as the applied Z-rotation $U(p_i, w_i)$. In subsequent sections, we will detail appending-matrix construction strategies along with the corresponding transversal physical Z-rotations tailored to satisfy these conditions for single-qubit and multi-controlled-Z rotations.

In summary, our appending framework provides a flexible means of extending a primary CSS code to support fault-tolerant implementations of desired logical dyadic Z-rotations, single-qubit as well as multi-controlled. For instance, one can realize both T and CS gates within the same code. As noted earlier, such a capability is particularly valuable when designing codes for quantum algorithms that require a diverse set of diagonal operations, as all gates are consolidated within a single code to maintain robust error correction. However, the cost we pay for this is the increased physical qubit overhead as the number of target logical gates grows.

5. Fault-tolerant Addressable Logical Single-Qubit Z-Rotations in CSS codes

For a non-negative integer ℓ, we begin this section by characterizing CSS codes that transversally support an addressable logical $R_Z\left(\frac{\pi}{2^\ell}\right)$ gate for an arbitrary but fixed addressability configuration. The following proposition is an immediate consequence of Theorem 5.

Proposition 8. *Let $(C_1, C_2)_{\mathsf{CSS}}$ be an $[[n, k]]$ CSS code. For any $A \in \mathbb{F}_2^k$, the code realizes the addressable logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ via a transversal physical Z-rotation $U(p, w)$ if and only if $p \geq \ell$ and, for all $x \in C_2$ and $a \in \mathbb{F}_2^k$, the following modular conditions hold:*

$$w \cdot x = 0 \pmod{2^{p+1}}, \qquad w \cdot y_a = 2^{p-\ell} w_H(a * A) \pmod{2^{p+1}},$$
$$w \cdot (x * y_a) = 0 \pmod{2^p}.$$

We now provide an alternative characterization of CSS codes capable of supporting the addressable logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$, decomposing the modular condition on the coset space C_1/C_2 into equivalent constraints on its basis vectors. Furthermore, we use this characterization to construct the appending matrices introduced in Section 4 to realize these gates. While Proposition 8 allows any $p \geq \ell$, for our purposes, it suffices to take $p = \ell$.

We begin with a technical lemma that reduces modular constraints over the entire code to equivalent conditions over its basis vectors.

Lemma 9. *Let $C \subseteq \mathbb{F}_2^n$ be a binary code with basis $\{x_1, \ldots, x_k\}$, and let $A \in \mathbb{F}_2^k$ be a fixed vector. For any $a \in \mathbb{F}_2^k$, define the corresponding codeword $x_a = \bigoplus_{i=1}^k a(i)x_i$. For any integer vector $w \in \mathbb{Z}^n$, the modular condition*

$$w \cdot x_a = w_H(a * A) \pmod{2^{\ell+1}} \tag{15}$$

holds for all $a \in \mathbb{F}_2^k$ if and only if the following conditions are satisfied for all $i \in [k]$ and ordered indices $1 \le i_1 < \cdots < i_j \le k$ with $j \in \{2, \ldots, \ell+1\}$:

$$w \cdot x_i = A(i) \pmod{2^{\ell+1}}, \tag{16}$$

$$w \cdot (x_{i_1} * \cdots * x_{i_j}) = 0 \pmod{2^{\ell-j+2}}. \tag{17}$$

This lemma is proved in Appendix E. Applying the lemma to Proposition 8 yields the following characterization.

Proposition 10. *Let $(C_1, C_2)_{\text{CSS}}$ be an $[[n, k]]$ CSS code, where $\{y_1, \ldots, y_k\}$ is a basis of C_1/C_2. For any $A \in \mathbb{F}_2^k$, the code realizes the addressable logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ via a transversal physical Z-rotation $U(\ell, w)$ if and only if for all $x \in C_2$, $a \in \mathbb{F}_2^k$, $i \in [k]$, and ordered indices $1 \le i_1 < \cdots < i_j \le k$ with $j \in \{2, \ldots, \ell+1\}$, the following modular equations hold:*

$$w \cdot x = 0 \pmod{2^{\ell+1}}, \qquad w \cdot (x * y_a) = 0 \pmod{2^\ell},$$
$$w \cdot y_i = A(i) \pmod{2^{\ell+1}}, \qquad w \cdot (y_{i_1} * \cdots * y_{i_j}) = 0 \pmod{2^{\ell-j+2}}.$$

5.1. **Appending Matrices for a Targeted Addressable Logical Gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$.** Given a primary $[[n', k']]$ CSS code $(C_1', C_2')_{\text{CSS}}$ and an arbitrary but fixed addressability vector $A \in \mathbb{F}_2^{k'}$, our objective is to extend the primary code to transversally realize the addressable logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$. We achieve this via the appending framework in Section 4, by deriving appending matrices tailored specifically to $R_Z\left(\frac{\pi}{2^\ell}\right)_A$. To construct these matrices, we leverage auxiliary CSS codes in which the physical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)^\dagger$ realizes the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$ (see Section 7).

To this end, let $(\widetilde{C}_1, \widetilde{C}_2)_{\text{CSS}}$ be an $[[\widetilde{n}, \widetilde{k}, \ge \widetilde{d}]]$ CSS code in which the physical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)^\dagger$ realizes the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$. We require $\widetilde{d}_Z \ge d_{\min}(\widetilde{C}_2^\perp) \ge \widetilde{d}$ so that all logical-Z operators (including Z-stabilizers) have weight at least $\widetilde{d}$. Following standard notation, let $G_{\widetilde{C}_2}$ and $G_{\widetilde{C}_1/\widetilde{C}_2}$ denote the generator matrices for the code $\widetilde{C}_2$ and the coset space $\widetilde{C}_1/\widetilde{C}_2$, respectively. Let $\widetilde{y}_1, \ldots, \widetilde{y}_{\widetilde{k}}$ denote the rows of $G_{\widetilde{C}_1/\widetilde{C}_2}$. By applying Proposition 10 with the negative all-ones vector $w = -\mathbf{1}_{\widetilde{n}}$ (since $R_Z\left(\frac{\pi}{2^\ell}\right)^\dagger = R_Z\left(\frac{\pi}{2^\ell}\right)^{-1}$), for all $\widetilde{x} \in \widetilde{C}_2$, $a \in \mathbb{F}_2^{\widetilde{k}}$, $i \in [\widetilde{k}]$, and ordered indices $1 \le i_1 < \cdots < i_j \le \widetilde{k}$ with $j \in \{2, \ldots, \ell+1\}$, the following conditions are satisfied:

$$w_H(\widetilde{x}) = 0 \pmod{2^{\ell+1}}, \qquad w_H(\widetilde{x} * \widetilde{y}_a) = 0 \pmod{2^\ell},$$
$$w_H(\widetilde{y}_i) = -1 \pmod{2^{\ell+1}}, \qquad w_H(\widetilde{y}_{i_1} * \cdots * \widetilde{y}_{i_j}) = 0 \pmod{2^{\ell-j+2}}, \tag{18}$$

where $\widetilde{y}_a = \bigoplus_{i=1}^{\widetilde{k}} a(i)\widetilde{y}_i$.

5.1.1. *Setting up the construction.* To align with the notation established in Section 4, assume that $M = 1$ and $U_{\mathrm{L}}^{(1)} = R_Z\left(\frac{\pi}{2^\ell}\right)_A$. For notational brevity, we temporarily suppress the superscript (1). Thus, G_1 and G_2 are the auxiliary matrices we need for our appending construction. The derived CSS code $(C_1, C_2)_{\mathsf{CSS}}$ is obtained from the generator matrices

$$G_{C_2} = \begin{bmatrix} G_{C_2'} & 0 \\ 0 & G_2 \end{bmatrix}, \qquad G_{C_1/C_2} = \begin{bmatrix} G_{C_1'/C_2'} & G_1 \end{bmatrix}, \qquad G_{C_1} = \begin{bmatrix} G_{C_1/C_2} \\ G_{C_2} \end{bmatrix}.$$

Let $y_1', \ldots, y_{k'}'$ denote the rows of $G_{C_1'/C_2'}$, and let $y_1, \ldots, y_{k'}$ denote the rows of the appended matrix G_1. The rows of G_{C_1/C_2} are $(y_1', y_1), \ldots, (y_{k'}', y_{k'})$. For any $a \in \mathbb{F}_2^{k'}$, the corresponding coset representative is given by (y_a', y_a), where $y_a' = \bigoplus_{i=1}^{k'} a(i) y_i'$ and $y_a = \bigoplus_{i=1}^{k'} a(i) y_i$.

Recall from the general appending framework that we aim to realize the addressable gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ via a transversal Z-rotation $U(\ell, w)$, where the vector w is supported solely on the appended coordinates. Let n'' denote the number of appended coordinates, or equivalently, the number of columns in G_1 (or G_2). By Proposition 10, the physical rotation $U(\ell, w)$ with $w = (\mathbf{0}_{n'}, -\mathbf{1}_{n''})$ realizes $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ within the code $(C_1, C_2)_{\mathsf{CSS}}$ if the following conditions hold for all $x \in$ rowspace(G_2), $a \in \mathbb{F}_2^{k'}$, $i \in [k']$, and ordered indices $1 \le i_1 < \cdots < i_j \le k'$ with $j \in \{2, \ldots, \ell+1\}$:

$$\begin{aligned} &w_H(x) = 0 \pmod{2^{\ell+1}}, && w_H(x * y_a) = 0 \pmod{2^\ell}, \\ &w_H(y_i) = -A(i) \pmod{2^{\ell+1}}, && w_H(y_{i_1} * \cdots * y_{i_j}) = 0 \pmod{2^{\ell-j+2}}. \end{aligned} \tag{19}$$

A comparison of (19) with (18) motivates us to extract vectors from the code $(\widetilde{C}_1, \widetilde{C}_2)_{\mathsf{CSS}}$ to construct G_1 and G_2. We first describe how to do this for the case when $|\mathrm{Supp}(A)| \le \widetilde{k}$, and subsequently show how this can be extended to the general case.

5.1.2. *The bounded support case:* $|\mathrm{Supp}(A)| \le \widetilde{k}$. For each $i \in [k']$, the construction must satisfy $w_H(y_i) = -A(i) \pmod{2^{\ell+1}}$. This motivates the following assignment strategy: when $A(i) = 0$, the row y_i is set to the zero vector, and when $A(i) = 1$, we populate the row with a vector $\widetilde{y}_j$. Crucially, the cross-weight condition $w_H(y_p * y_q) = 0 \pmod{2^\ell}$ strictly prohibits reusing the same vector $\widetilde{y}_j$ for distinct indices $p \neq q$ with $A(p) = A(q) = 1$. Consequently, each non-zero entry in A demands a unique $\widetilde{y}_j$. The assumption $|\mathrm{Supp}(A)| \le \widetilde{k}$ guarantees a sufficient supply of these distinct vectors, ensuring the assignment is realizable without repetition.

To simplify the presentation of the construction, we assume that $A(i) = 1$ if and only if $i \in [\mathsf{s}]$, where $\mathsf{s} := |\mathrm{Supp}(A)| \le \widetilde{k}$. We define the rows $y_1, \ldots, y_{k'}$ of G_1 as follows: we set $y_i = \widetilde{y}_i$ for target rows $i \in [\mathsf{s}]$, and set $y_i = 0$ for all remaining rows $\mathsf{s}+1 \le i \le k'$.

Furthermore, setting $G_2 = G_{\widetilde{C}_2}$ satisfies all required modular equations to realize the logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ by applying physical $U(\ell, w)$ with $w = (\mathbf{0}_{n'}, -\mathbf{1}_{\widetilde{n}})$. Since $d_{\min}(\widetilde{C}_2^\perp) \ge \widetilde{d}$, by Lemma 7, the distance of the final code $(C_1, C_2)_{\mathsf{CSS}}$ is at least $\min\{d_X, d_Z\}$, where $d_X \ge d'$ and $d_Z \ge \min\{d_Z', \widetilde{d}\}$.

5.1.3. *Removing the restriction on* $|\mathrm{Supp}(A)|$. A limitation of the construction described in Section 5.1.2 is that it does not directly scale: if $|\mathrm{Supp}(A)| > \widetilde{k}$, we exhaust the available supply of coset vectors $\widetilde{y}_j$, and reusing them violates the required modular constraints (specifically, $w_H(y_p * y_q) = 0 \pmod{2^\ell}$). To overcome this, we decompose the logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ into $g := \lceil |\mathrm{Supp}(A)|/\widetilde{k} \rceil$ addressable gates $R_Z\left(\frac{\pi}{2^\ell}\right)_{A_1}, \ldots, R_Z\left(\frac{\pi}{2^\ell}\right)_{A_g}$ such that each of the first $g-1$ address vectors A_i ($i = 1, 2, \ldots, g-1$) has support size equal to $\widetilde{k}$, and $|\mathrm{Supp}(A_g)| \leq \widetilde{k}$. To be precise, again setting $\mathsf{s} = |\mathrm{Supp}(A)|$, and letting $j_1, j_2, \ldots, j_{\mathsf{s}}$ denote the indices in $\mathrm{Supp}(A)$ listed in increasing order, we define the address vectors for $1 \leq i \leq g-1$ by $A_i = \left(\mathbf{0}_{j_{(i-1)\widetilde{k}}}, A\left(j_{(i-1)\widetilde{k}} + 1 : j_{i\widetilde{k}}\right), \mathbf{0}_{k' - j_{i\widetilde{k}}}\right)$, and the final vector by $A_g = \left(\mathbf{0}_{j_{(g-1)\widetilde{k}}}, A\left(j_{(g-1)\widetilde{k}} + 1 : k'\right)\right)$, where $j_0 := 0$ and $A(\ell_1 : \ell_2) := (A(\ell_1), A(\ell_1 + 1), \ldots, A(\ell_2))$. In other words, A_1 is determined by the prefix of A up to and including the $\widetilde{k}$-th 1, A_2 by the next sub-block of A up to the $2\widetilde{k}$-th 1, and so on up to A_{g-1}, with A_g capturing the remaining suffix. Altogether, we have $R_Z\left(\frac{\pi}{2^\ell}\right)_A = \prod_{i=1}^{g} R_Z\left(\frac{\pi}{2^\ell}\right)_{A_i}$.

We now instantiate the general appending framework from Section 4 with $M = g$, setting $U_{\mathrm{L}}^{(i)} = R_Z\left(\frac{\pi}{2^\ell}\right)_{A_i}$ for each $i \in [g]$. Because $|\mathrm{Supp}(A_i)| \leq \widetilde{k}$, we can leverage the strategy in Section 5.1.2 to construct the corresponding appending matrices. Thus, for each $i \in [g]$, a transversal Z-rotation $U(\ell, w_i)$ realizes the logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_{A_i}$, where w_i is supported strictly on the appended coordinates corresponding to $R_Z\left(\frac{\pi}{2^\ell}\right)_{A_i}$:

$$w_i = (\mathbf{0}_{n' + (i-1)\widetilde{n}}, -\mathbf{1}_{\widetilde{n}}, \mathbf{0}_{(g-i)\widetilde{n}}).$$

Consequently, the product of these physical rotations, $\prod_{i=1}^{g} U(\ell, w_i) = U\left(\ell, \sum_{i=1}^{g} w_i\right)$, realizes the target logical gate $\prod_{i=1}^{g} R_Z\left(\frac{\pi}{2^\ell}\right)_{A_i} = R_Z\left(\frac{\pi}{2^\ell}\right)_A$. Since $\sum_{i=1}^{g} w_i = (\mathbf{0}_{n'}, -\mathbf{1}_{\widetilde{n}g})$, we conclude that the target logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ is realized within the CSS code $(C_1, C_2)_{\mathsf{CSS}}$ by applying the transversal Z-rotation $U(\ell, w)$, with $w = (\mathbf{0}_{n'}, -\mathbf{1}_{\widetilde{n}g})$, to the physical qubits of the code. Finally, Lemma 7 guarantees the resulting code $(C_1, C_2)_{\mathsf{CSS}}$ has distance at least $\min\{d_X, d_Z\}$, with $d_X \geq d'$ and $d_Z \geq \min\{d'_Z, \widetilde{d}\}$.

5.1.4. *Summary.* We now summarize the appending-matrix construction. Fix a non-negative integer ℓ and an $[[\widetilde{n}, \widetilde{k}, \geq \widetilde{d}]]$ auxiliary code wherein the physical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)^\dagger$ realizes the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$. Given a primary CSS code $(C'_1, C'_2)_{\mathsf{CSS}}$ and a target address vector $A \in \mathbb{F}_2^{k'}$, we perform the following steps to obtain the derived code:

(1) Determine the number of appending matrices $M = \lceil |\mathrm{Supp}(A)|/\widetilde{k} \rceil$, and define the matrices $G_1^{(j)} \in \mathbb{F}_2^{k' \times \widetilde{n}}$ and $G_2^{(j)} \in \mathbb{F}_2^{k_2 \times \widetilde{n}}$ for $j \in [M]$, where $k_2 := \dim(\widetilde{C}_2)$.

(2) Let $\mathsf{s} = |\mathrm{Supp}(A)|$, and let $1 \leq i_1 < \cdots < i_{\mathsf{s}} \leq k'$ denote the ordered indices in $\mathrm{Supp}(A)$. Setting $i_0 := 0$, we decompose A into M address vectors $A_1, \ldots, A_M$, where

$$A_j = \left(\mathbf{0}_{i_{(j-1)\widetilde{k}}}, A\left(i_{(j-1)\widetilde{k}} + 1 : i_{j\widetilde{k}}\right), \mathbf{0}_{k' - i_{j\widetilde{k}}}\right), \quad \text{for } j \in [M-1] \text{ and}$$

$$A_M = \left(\mathbf{0}_{i_{(M-1)\widetilde{k}}}, A\left(i_{(M-1)\widetilde{k}} + 1 : k'\right)\right)$$

(3) For each $j \in [M]$:

(a) Let $\mathsf{s}_j = |\mathrm{Supp}(A_j)|$, and let $1 \le p_1 < \cdots < p_{\mathsf{s}_j}$ denote the indices in $\mathrm{Supp}(A_j)$. We define the rows $y_1^{(j)}, \ldots, y_{k'}^{(j)}$ of $G_1^{(j)}$ for each $h \in [k']$ as

$$y_h^{(j)} = \begin{cases} \widetilde{y}_q, & \text{if } h = p_q \text{ for some } q \in [\mathsf{s}_j], \\ \mathbf{0}_{\widetilde{n}}, & \text{otherwise.} \end{cases}$$

(b) Set $G_2^{(j)} = G_{\widetilde{C}_2}$.

(4) The generator matrices of the final derived CSS code $(C_1, C_2)_{\mathrm{CSS}}$ are given by

$$G_{C_2} = \begin{bmatrix} G_{C_2'} & 0 & 0 & \cdots & 0 \\ 0 & G_2^{(1)} & 0 & \cdots & 0 \\ 0 & 0 & G_2^{(2)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & G_2^{(M)} \end{bmatrix} \quad \text{and} \quad G_{C_1/C_2} = \begin{bmatrix} G_{C_1'/C_2'} & G_1^{(1)} & G_1^{(2)} & \cdots & G_1^{(M)} \end{bmatrix}.$$

The above procedure yields a derived CSS code with parameters $[[n' + \widetilde{n}M, k', \geq \min\{d', \widetilde{d}\}]]$ in which the transversal physical Z-rotation $U(\ell, w)$ with $w = (\mathbf{0}_{n'}, -\mathbf{1}_{\widetilde{n}M})$ realizes the target logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$.

The overall assignment strategy guarantees that no coset vector $\widetilde{y}_j$ is repeated within any single block $G_1^{(i)}$. The rationale behind partitioning $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ thus becomes apparent: if y_p and y_q are any two distinct non-zero rows of $\begin{bmatrix} G_1^{(1)} & \cdots & G_1^{(M)} \end{bmatrix}$, then either $w_H(y_p * y_q) = w_H(\widetilde{y}_{j_1} * \widetilde{y}_{j_2})$ for some $j_1 \neq j_2$ (this happens if the two rows pass through the same block $G_1^{(i)}$) or $y_p * y_q = 0$. In either case, $w_H(y_p * y_q) = 0 \pmod{2^\ell}$.

We next illustrate the mechanics of the appending construction with a concrete example.

Example 2. Let the primary CSS component of our construction be the order-4 quantum Hamming code [68], denoted $(C_1', C_2')_{\mathrm{CSS}}$, which achieves parameters $[[n' = 15, k' = 7, d' = 3]]$. Let the vectors $y_1', \ldots, y_7'$ denote the rows of the coset generator matrix $G_{C_1'/C_2'}$. Given a target addressability vector $A \in \mathbb{F}_2^7$ defined by non-zero coordinates $A(1) = A(3) = A(6) = 1$, our goal is to construct a derived CSS code encoding $k' = 7$ logical qubits that realizes the addressable logical gate T_A, where $T = R_Z\left(\frac{\pi}{4}\right)$.

Our second component is an auxiliary CSS code, $(\widetilde{C}_1, \widetilde{C}_2)_{\mathrm{CSS}}$ wherein the physical transversal $T^\dagger$ realizes the logical transversal T. We fulfill this using a $[[\widetilde{n} = 14, \widetilde{k} = 2, \widetilde{d} \geq 2]]$ code derived by puncturing two coordinates of the Reed–Muller code $\mathrm{RM}(1, 4)$. While the existence of this code is well known, we will detail its construction in the Section 7. Let the vectors $\widetilde{y}_1, \widetilde{y}_2$ denote the rows of the coset space generator matrix $G_{\widetilde{C}_1/\widetilde{C}_2}$.

We now commence the construction of the appending matrices. Since $|\mathrm{Supp}(A)| \geq \widetilde{k}$, we decompose T_A into $g = \lceil |\mathrm{Supp}(A)|/\widetilde{k} \rceil = 2$ gates T_{A_1} and T_{A_2}, where

$$A_1 = (1,0,1,0,0,0,0) \quad \text{and} \quad A_2 = (0,0,0,0,0,1,0).$$

Setting $U_{\mathrm{L}}^{(1)} = T_{A_1}$ and $U_{\mathrm{L}}^{(2)} = T_{A_2}$, the appending matrices $G_1^{(1)}$ and $G_1^{(2)}$ in $\mathbb{F}_2^{7\times 14}$ are constructed as follows:

$$G_1^{(1)} = \begin{bmatrix} \widetilde{y}_1 \\ 0 \\ \widetilde{y}_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad G_1^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \widetilde{y}_1 \\ 0 \end{bmatrix},$$

while the accompanying matrices $G_2^{(1)}$ and $G_2^{(2)}$ are given by

$$G_2^{(1)} = G_2^{(2)} = G_{\widetilde{C}_2}.$$

Let $(C_1, C_2)_{\mathrm{CSS}}$ denote the resultant CSS code obtained by integrating these appending matrices into the primary code. The generator matrix of the subcode C_2 is given by

$$G_{C_2} = \begin{bmatrix} G_{C_2'} & 0 & 0 \\ 0 & G_2^{(1)} & 0 \\ 0 & 0 & G_2^{(2)} \end{bmatrix} = \begin{bmatrix} G_{C_2'} & 0 & 0 \\ 0 & G_{\widetilde{C}_2} & 0 \\ 0 & 0 & G_{\widetilde{C}_2} \end{bmatrix},$$

and the generator matrix for the coset space C_1/C_2 is given by

$$G_{C_1/C_2} = \begin{bmatrix} G_{C_1'/C_2'} & G_1^{(1)} & G_1^{(2)} \end{bmatrix} = \left[\begin{array}{c|c|c} y_1' & \widetilde{y}_1 & 0 \\ y_2' & 0 & 0 \\ y_3' & \widetilde{y}_2 & 0 \\ y_4' & 0 & 0 \\ y_5' & 0 & 0 \\ y_6' & 0 & \widetilde{y}_1 \\ y_7' & 0 & 0 \end{array}\right].$$

This completes our appending construction. By invoking Lemma 7, we deduce that $(C_1, C_2)_{\mathrm{CSS}}$ is a $[[43, 7, \geq 2]]$ code. Applying the transversal Z-rotation $U(2, w_1)$ with $w_1 = (\mathbf{0}_{15}, -\mathbf{1}_{14}, \mathbf{0}_{14})$ realizes logical T_{A_1}, while $U(2, w_2)$ with $w_2 = (\mathbf{0}_{15}, \mathbf{0}_{14}, -\mathbf{1}_{14})$ realizes logical T_{A_2}. Consequently, the product of these physical rotations, $U(2, w_1)U(2, w_2) = U(2, w)$ with combined vector $w = (\mathbf{0}_{15}, -\mathbf{1}_{14}, -\mathbf{1}_{14})$, successfully realizes the target logical gate $T_{A_1}T_{A_2} = T_A$.

We end this section by pointing out that the methodology herein enables us to construct CSS codes that admit transversal realizations of all possible addressable logical $R_Z\left(\frac{\pi}{2^\ell}\right)$ gates, for some

fixed $\ell \geq 0$. This is based on the observation that to do this, it suffices to construct CSS codes that admit transversal realizations of $R_Z\left(\frac{\pi}{2^\ell}\right)$ on each logical qubit independently. Indeed, suppose that for each i, a transversal Z-rotation $U(\ell, w_i)$ realizes the logical $R_Z\left(\frac{\pi}{2^\ell}\right)$ gate on the i-th logical qubit. Then, for any target address vector A, the composite gate $\prod_i U(\ell, w_i)^{A(i)}$ realizes the addressable logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$.

The construction of such CSS codes is straightforward: we simply invoke the appending framework of Section 4 with $M = k'$ and set $U_{\mathrm{L}}^{(i)} = R_Z\left(\frac{\pi}{2^\ell}\right)_{e_i}$. The strategy described in Section 5.1.2 tells us how to choose appending matrices for single-qubit Z-rotations with address vectors supported on a single coordinate.

5.2. **Relaxed Modular Equations to Realize Addressable Logical Single-Qubit Z-Rotations.** The system of modular equations in Proposition 10 can be further relaxed. Specifically, weakening the modulo-$2^{\ell+1}$ constraint $w \cdot y_i = A(i) \pmod{2^{\ell+1}}$ to make it a modulo-2^ℓ constraint still allows the code to transversally realize the logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$, as we now demonstrate.

Using Lemma 17 in Appendix B, we observe that any vector w satisfying the relaxed conditions can be systematically refined into a vector w' that satisfies the original modular equations, thereby realizing the target gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ via $U(\ell, w')$. Applying Lemma 17 to Proposition 10 yields the following alternative characterization:

Proposition 11. *Let $(C_1, C_2)_{\mathrm{CSS}}$ be an $[[n, k]]$ CSS code, where $\{x_1, \ldots, x_{k_2}\}$ and $\{y_1, \ldots, y_k\}$ are bases for C_2 and C_1/C_2, respectively. For any $A \in \mathbb{F}_2^k$, the code realizes the addressable logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ via a transversal physical Z-rotation $U(\ell, w')$ for some $w' \in \mathbb{Z}^n$ if and only if there exists a vector $w \in \mathbb{Z}^n$ such that for all $a \in \mathbb{F}_2^k$, $i \in [k_2]$, ordered indices $1 \leq i_1 < \cdots < i_j \leq k_2$ with $j \in \{2, \ldots, \ell+1\}$, $p \in [k]$, and ordered indices $1 \leq p_1 < \cdots < p_q \leq k$ with $q \in \{2, \ldots, k\}$, the following conditions hold:*

$$w \cdot x_i = 0 \pmod{2^\ell}, \qquad w \cdot (x_{i_1} * \cdots * x_{i_j}) = 0 \pmod{2^{\ell-j+2}}, \qquad w \cdot (x * y_a) = 0 \pmod{2^\ell},$$

$$w \cdot y_p = A(p) \pmod{2^\ell}, \qquad w \cdot (y_{p_1} * \cdots * y_{p_q}) = 0 \pmod{2^{\ell-q+2}}.$$

The vector w in Proposition 11 need not satisfy the original characterization in Proposition 10 required to realize the addressable logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ directly. Instead, by Lemma 17, we can map the physical rotation $U(\ell, w)$ to a refined rotation $U(\ell, w')$, where the modified vector $w' = w - 2^\ell w_1 - 2^\ell w_2$ exactly satisfies the original characterization in Proposition 10. Here, following the proof of Lemma 17, the correction vector w_1 is obtained when lifting the conditions on $w \cdot x_i$ from $0 \pmod{2^\ell}$ to $0 \pmod{2^{\ell+1}}$, and w_2 is obtained when lifting the conditions on $w \cdot y_p$ from $A(p) \pmod{2^\ell}$ to $A(p) \pmod{2^{\ell+1}}$.

Because $U(\ell, w')$ differs from $U(\ell, w)$ only by the application of physical Pauli Z gates, the relaxed conditions can be interpreted as characterizing all CSS codes capable of realizing the addressable logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_A$ via transversal physical Z-rotations, up to the application of Pauli Z corrections. In particular, the process of lifting the modular conditions using Lemma 17 establishes the sufficiency of Proposition 11. Conversely, the necessity follows immediately, as any

vector satisfying the strict characterization of Proposition 10 trivially satisfies the relaxed modular conditions.

6. Fault-tolerant Addressable Logical $C\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ Gates in CSS Codes

In this section, we utilize Proposition 6 to derive an alternative characterization of CSS codes that transversally realize an addressable logical $C\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ gate, which motivates our appending-matrix construction for 1-controlled-Z rotations. Although Proposition 6 allows any $p \geq \ell + 1$, we restrict our attention to $p = \ell + 1$.

While the following characterization and appending-matrix construction extend to general multi-controlled-Z rotations, the associated technical details can obscure the core ideas. Therefore, we focus here on 1-controlled-Z rotations and defer a self-contained treatment of the general case to Appendix A.

Lemma 13 in Appendix A extends Lemma 9 from binary vectors $A \in \mathbb{F}_2^k$ to binary tensors $A \in \mathbb{F}_2^{k\times\cdots\times k}$. Applying it to Proposition 6 for 1-controlled-Z rotations directly yields the following result.

Proposition 12. *Let $(C_1, C_2)_{\text{CSS}}$ be an $[[n, k]]$ CSS code, where $\{y_1, \ldots, y_k\}$ is a basis of C_1/C_2. For any matrix $A \in \mathbb{F}_2^{k\times k}$, the code realizes the addressable logical gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ via a transversal physical Z-rotation $U(\ell + 1, w)$ if and only if for all $x \in C_2$, $a \in \mathbb{F}_2^k$, and ordered indices $1 \leq i_1 < \cdots < i_j \leq k$ with $j \in [\ell + 2]$, the following conditions hold:*

$$w \cdot x = 0 \pmod{2^{\ell+2}}, \qquad w \cdot (x * y_a) = 0 \pmod{2^{\ell+1}},$$

$$w \cdot \left(y_{i_1} * \cdots * y_{i_j}\right) = \begin{cases} -A(i_1, i_2) \pmod{2^{\ell+1}}, & \text{if } j = 2, \\ 0 \pmod{2^{\ell-j+3}}, & \text{otherwise.} \end{cases}$$

6.1. Appending Matrices for a Targeted Addressable Logical gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$.

Given a primary $[[n', k']]$ CSS code $(C_1', C_2')_{\text{CSS}}$ and a target address matrix $A \in \mathbb{F}_2^{k'\times k'}$, we utilize the appending framework from Section 4 to construct an extended code that transversally realizes the addressable logical gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$. We derive the required appending matrices using the auxiliary $[[\widetilde{n}, \widetilde{k}, \widetilde{d}]]$ CSS code $(\widetilde{C}_1, \widetilde{C}_2)_{\text{CSS}}$ from Section 5.1, wherein the physical transversal $R_Z\left(\frac{\pi}{2^{\ell+1}}\right)^\dagger$ realizes the logical transversal $R_Z\left(\frac{\pi}{2^{\ell+1}}\right)$, and $\widetilde{d}_Z \geq d_{\min}(\widetilde{C}_2^\perp) \geq \widetilde{d}$.

Recall from Section 5.1 that if $\widetilde{y}_1, \ldots, \widetilde{y}_{\widetilde{k}}$ are the rows of the coset generator matrix $G_{\widetilde{C}_1/\widetilde{C}_2}$, the following hold for all $\widetilde{x} \in \widetilde{C}_2$, $a \in \mathbb{F}_2^{\widetilde{k}}$, $i \in [\widetilde{k}]$, and ordered indices $1 \leq i_1 < \cdots < i_j \leq \widetilde{k}$ with $j \in \{2, \ldots, \ell + 2\}$:

$$w_H(\widetilde{x}) = 0 \pmod{2^{\ell+2}}, \qquad w_H(\widetilde{x} * \widetilde{y}_a) = 0 \pmod{2^{\ell+1}},$$

$$w_H(\widetilde{y}_i) = -1 \pmod{2^{\ell+2}}, \qquad w_H(\widetilde{y}_{i_1} * \cdots * \widetilde{y}_{i_j}) = 0 \pmod{2^{\ell-j+3}}. \tag{20}$$

Our construction requires the following properties of the rows of $G_{\widetilde{C}_1/\widetilde{C}_2}$ established in Proposition 15:

(1)

$$w_H\left(\bigoplus_{i=1}^{\widetilde{k}}\widetilde{y}_i\right) = -\widetilde{k} \pmod{2^{\ell+2}}. \tag{21}$$

(2) For any index $p \in [\widetilde{k}]$,

$$w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}}\widetilde{y}_i\right) * \widetilde{y}_p\right) = -1 \pmod{2^{\ell+1}}. \tag{22}$$

(3) For any integer $q \geq 2$ and ordered sequence of indices $1 \leq p_1 < \cdots < p_q \leq \widetilde{k}$,

$$w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}}\widetilde{y}_i\right) * \left(\widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}\right)\right) = 0 \pmod{2^{\ell-q+2}}. \tag{23}$$

6.1.1. *Reduction to a simpler special case.* We first decompose the target gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ by sequentially ordering its control qubits. Let $c_1 < \cdots < c_\kappa$ denote the ordered indices in $[k'-1]$ corresponding to active control qubits—that is, indices i for which there exists at least one target qubit $t \in \{i+1, \ldots, k'\}$ such that $A(i,t) = 1$. We factor $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ into κ addressed gates $C_{A_1}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right), \ldots, C_{A_\kappa}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$. Each matrix A_j isolates the interactions controlled by c_j by inheriting the c_j-th row slice of A and vanishing elsewhere:

$$A_j(p,q) := \begin{cases} A(c_j,q), & \text{if } p = c_j \text{ and } q \in \{c_j+1, \ldots, k'\}, \\ 0, & \text{otherwise.} \end{cases}$$

If the appending-matrix construction is established for each factor $C_{A_j}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$, the appending framework guarantees that each gate $C_{A_j}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ is realized by some transversal Z-rotation $U(\ell+1, w_j)$ in the final extended code. The composition $\prod_{j=1}^{\kappa} U(\ell+1, w_j)$ then realizes the full target gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$. Consequently, without loss of generality, we assume that the first row is the only non-zero row of A—meaning the logical $C\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ gates act exclusively with the first logical qubit as the control qubit and a subset of the remaining qubits as targets.

Furthermore, we restrict the construction to target matrices satisfying $|\mathrm{Supp}(A(1, 2:k'))| \leq \widetilde{k}$, where $A(p, q:r) := (A(p,q), A(p,q+1), \ldots, A(p,r))$ for $p \in [k']$ and $1 \leq q \leq r \leq k'$. If the target support exceeds $\widetilde{k}$, the flexibility of the appending framework allows us to partition the addressable gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ into $g = \lceil |\mathrm{Supp}(A(1, 2:k'))| / \widetilde{k} \rceil$ factors, each acting from control qubit 1 onto at most $\widetilde{k}$ targets in $\{2, \ldots, k'\}$. Constructing appending matrices independently for each factor and composing their corresponding physical Z-rotations realizes the target gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ within the final extended code.

6.1.2. *The appending-matrix construction in the special case of* $|\mathrm{Supp}(A(1,2:k'))| \leq \widetilde{k}$. To simplify the presentation of the construction, we further assume that $A(i,j) = 1$ if and only if $i = 1$ and $j \in \{2,3,\ldots,\mathsf{s}+1\}$, where $\mathsf{s} := |\mathrm{Supp}(A(1,2:k'))| \leq \widetilde{k}$. This effectively designates $\{2,\ldots,\mathsf{s}+1\}$ as the set of all target qubits. Again, recall that we are given a primary $[[n',k']]$ CSS code $(C_1', C_2')_{\mathsf{CSS}}$, and that we will obtain appending matrices G_1 and G_2 using an auxiliary $[[\widetilde{n},\widetilde{k},\widetilde{d}]]$ CSS code $(\widetilde{C}_1,\widetilde{C}_2)_{\mathsf{CSS}}$ for which the modular equations in (20) hold. We define the rows $y_1,\ldots,y_{k'}$ of G_1 as follows: we set $y_1 = \bigoplus_{p=1}^{\widetilde{k}} \widetilde{y}_p$ for the control qubit row, assign $y_j = \widetilde{y}_{j-1}$ for target qubits $2 \leq j \leq \mathsf{s}+1$, and set $y_j = 0$ for all remaining rows $\mathsf{s}+1 < j \leq k'$.

Next, we set $G_2 = G_{\widetilde{C}_2}$, and let $(\widehat{C}_1,\widehat{C}_2)_{\mathsf{CSS}}$ denote the CSS code obtained by applying the appending-matrix construction to $(C_1', C_2')_{\mathsf{CSS}}$.

6.1.3. $U(\ell+1,w)$ *is a logical operator for* $(\widehat{C}_1,\widehat{C}_2)_{\mathsf{CSS}}$. Consider the transversal Z-rotation $U(\ell+1,w)$ with $w = (\mathbf{0}_{n'},\mathbf{1}_{\widetilde{n}})$. By construction, $G_2 = G_{\widetilde{C}_2}$ and the rows of G_1 are drawn from $\widetilde{C}_1/\widetilde{C}_2$. Consequently, for any $x = (x',\widetilde{x}) \in \widehat{C}_2$ and $y_a = (y',\widetilde{y}) \in \widehat{C}_1/\widehat{C}_2$ (with $a \in \mathbb{F}_2^{k'}$), we have $\widetilde{x} \in \widetilde{C}_2$ and $\widetilde{y} \in \widetilde{C}_1/\widetilde{C}_2$. By (20), this guarantees

$$w \cdot x = 0 \pmod{2^{\ell+2}} \quad \text{and} \quad w \cdot (x * y_a) = 0 \pmod{2^{\ell+1}}.$$

Thus, Remark 4 establishes $U(\ell+1,w)$ as a logical operator on $(\widehat{C}_1,\widehat{C}_2)_{\mathsf{CSS}}$.

6.1.4. $U(\ell+1,w)$ *realizes* $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ *up to a logical phase.* For any logical state $|a\rangle$ with $a \in \mathbb{F}_2^{k'}$, the action of $U(\ell+1,w)$ is given by

$$\begin{aligned} U(\ell+1,w)\,|a\rangle_L &= \exp\left\{ \iota \frac{\pi}{2^{\ell+1}} \left(w \cdot \left(\bigoplus_{i=1}^{k'} a(i) y_i \right) \right) \right\} |a\rangle_L \\ &= \exp\left\{ \iota \frac{\pi}{2^{\ell+1}} \left(\sum_{i=1}^{k'} (-2)^{i-1} \left(\sum_{1 \leq j_1 < \cdots < j_i \leq k'} \left(\prod_{p=1}^{i} a(j_p) \right) w_H\left(y_{j_1} * \cdots * y_{j_i} \right) \right) \right) \right\} |a\rangle_L . \end{aligned}$$

Separating the linear and quadratic terms, we can express this action as

$$\begin{aligned} U(\ell+1,w)\,|a\rangle_L = \exp\Bigg\{ \iota \frac{\pi}{2^{\ell+1}} \Bigg(& \sum_{i=1}^{k'} a(i) w_H(y_i) - 2 \sum_{1 \leq i < j \leq k'} a(i)a(j) w_H(y_i * y_j) \\ & + \sum_{i=3}^{k'} (-2)^{i-1} \left(\sum_{1 \leq j_1 < \cdots < j_i \leq k'} \left(\prod_{p=1}^{i} a(j_p) \right) w_H\left(y_{j_1} * \cdots * y_{j_i} \right) \right) \Bigg) \Bigg\} |a\rangle_L . \end{aligned}$$

We next show that the t-fold Schur products for $t > 2$ vanish in the phase exponent, while the 2-fold products exactly realize the target gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$. The 1-fold terms (the linear terms), however, introduce an extraneous logical phase, which we will eliminate in the next subsection.

To see why the t-fold Schur products for $t > 2$ vanish, first consider any ordered sequence of $t > 2$ target qubit indices $2 \leq p_1 < \cdots < p_t \leq \mathsf{s}+1$. Condition (20) implies

$$w_H(y_{p_1} * \cdots * y_{p_t}) = w_H(\widetilde{y}_{p_1-1} * \cdots * \widetilde{y}_{p_t-1}) = 0 \pmod{2^{\ell-t+3}}.$$

On the other hand, if the Schur product combines the control qubit index with $t \geq 2$ target qubit indices, (23) yields

$$w_H(y_1 * y_{p_1} * \cdots * y_{p_t}) = w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) * \widetilde{y}_{p_1-1} * \cdots * \widetilde{y}_{p_t-1}\right) = 0 \pmod{2^{\ell-t+2}}.$$

Because only the first $\mathsf{s}+1$ rows of G_1 are non-zero, the preceding modular equations guarantee that all t-fold Schur products for $t > 2$ vanish in the phase exponent. Consequently, the action of the physical rotation $U(\ell+1, w)$ reduces to the linear and quadratic terms:

$$U(\ell+1,w)\,|a\rangle_L = \exp\left\{\iota\frac{\pi}{2^{\ell+1}}\left(\sum_{i=1}^{k'} a(i)w_H(y_i) - 2\sum_{1\leq i<j\leq k'} a(i)a(j)w_H(y_i * y_j)\right)\right\}|a\rangle_L\,.$$

We now evaluate the 2-fold Schur products of non-zero rows of G_1 in the phase exponent. For any target qubit indices $2 \leq p < q \leq \mathsf{s}+1$, (20) gives

$$w_H(y_p * y_q) = w_H(\widetilde{y}_{p-1} * \widetilde{y}_{q-1}) = 0 \pmod{2^{\ell+1}}.$$

If the product instead involves the control qubit and any target qubit $2 \leq p \leq \mathsf{s}+1$, (22) yields

$$w_H(y_1 * y_p) = w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) * \widetilde{y}_{p-1}\right) = -1 \pmod{2^{\ell+1}}.$$

Substituting these evaluations into the quadratic terms, and recalling that $A(i,j) = 1$ if and only if for $i = 1$ and $j \in \{2, \ldots, \mathsf{s}+1\}$, we see that the logical action of the transversal physical rotation $U(\ell+1, w)$ further reduces to

$$\begin{aligned} U(\ell+1,w)\,|a\rangle_L &= \exp\left\{\iota\frac{\pi}{2^{\ell+1}}\left(\sum_{i=1}^{k'} a(i)w_H(y_i) + 2\sum_{1\leq i<j\leq k'} a(i)a(j)A(i,j)\right)\right\}|a\rangle_L \\ &= \exp\left\{\iota\frac{\pi}{2^{\ell+1}}\left(\sum_{i=1}^{k'} a(i)w_H(y_i)\right)\right\} C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)|a\rangle_L\,. \end{aligned}$$

Thus, we have successfully realized the target gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ up to a logical phase.

6.1.5. *Eliminating the residual logical phase.* By (20) and (21), we have $w_H(\widetilde{y}_i) = -1 \pmod{2^{\ell+2}}$ and $w_H\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) = -\widetilde{k} \pmod{2^{\ell+2}}$. Substituting these values allows us to rewrite the logical action as

$$U(\ell+1,w)\,|a\rangle_L = \exp\left\{\iota\frac{\pi}{2^{\ell+1}}\left(-\widetilde{k}\sum_{i=1}^{k'} a(i)A_1(i) - \sum_{i=1}^{k'} a(i)A_2(i)\right)\right\} C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)|a\rangle_L\,,$$

where $A_1 = (1, \mathbf{0}_{k'-1})$ and $A_2 = (0, \mathbf{1}_{\mathsf{s}}, \mathbf{0}_{k'-\mathsf{s}-1})$. We can thus explicitly identify the unwanted residual phase as the action of the logical Z-rotations $\left(R_Z\left(\frac{\pi}{2^{\ell+1}}\right)^\dagger_{A_1}\right)^{\widetilde{k}}$ and $R_Z\left(\frac{\pi}{2^{\ell+1}}\right)^\dagger_{A_2}$.

To cancel this unwanted logical phase, we append auxiliary matrices to $(\widehat{C}_1, \widehat{C}_2)_{\text{CSS}}$ that realize the inverse logical Z-rotations $\left(R_Z\left(\frac{\pi}{2^{\ell+1}}\right)_{A_1}\right)^{\widetilde{k}}$ and $R_Z\left(\frac{\pi}{2^{\ell+1}}\right)_{A_2}$. This yields the final derived CSS code $(C_1, C_2)_{\text{CSS}}$, which transversally realizes the exact target gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$. Note that the appending matrix required for the rotation $\left(R_Z\left(\frac{\pi}{2^{\ell+1}}\right)_{A_1}\right)^{\widetilde{k}}$ is the same as that of the rotation $R_Z\left(\frac{\pi}{2^\ell}\right)_{A_1}$. This is because if a transversal physical Z-rotation U_{P} realizes $R_Z\left(\frac{\pi}{2^\ell}\right)_{A_1}$, then $U_{\text{P}}^{\widetilde{k}}$ realizes $\left(R_Z\left(\frac{\pi}{2^\ell}\right)_{A_1}\right)^{\widetilde{k}}$.

We note here that although not explicitly invoked, Proposition 12 motivates our construction. We define the control qubit indexed row as the sum $\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i$ and the target qubit indexed rows as the vectors $\widetilde{y}_j$ specifically to satisfy the modular equation $w \cdot (y_{i_1} * y_{i_2}) = -A(i_1, i_2) \pmod{2^{\ell+1}}$. The intermediate code $(\widehat{C}_1, \widehat{C}_2)_{\text{CSS}}$ satisfies all requirements of the proposition except $w \cdot y_{i_1} = 0 \pmod{2^{\ell+2}}$. Appending auxiliary matrices to this intermediate code to cancel the residual logical phase precisely resolves this modular equation, ensuring the final derived code fully satisfies Proposition 12.

Finally, observe that any code realizing the logical transversal $R_Z\left(\frac{\pi}{2^{\ell+1}}\right)$ can also realize the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$. Consequently, for the same blocklength $\widetilde{n}$, one can always find codes that realize the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$ with logical dimension and minimum distance at least $\widetilde{k}$ and $\widetilde{d}$, respectively. This ensures that the rotations $R_Z\left(\frac{\pi}{2^{\ell+1}}\right)_{A_1}$ and $R_Z\left(\frac{\pi}{2^{\ell+1}}\right)_{A_2}$ trivially satisfy the bounded support condition of Section 5.1.2, so that the total appended blocklength in the resulting code will be $3\widetilde{n}$.

6.1.6. *Summary.* We now summarize the appending-matrix construction for 1-controlled-Z rotations. Fix a non-negative integer ℓ and an $[[\widetilde{n}, \widetilde{k}, \geq \widetilde{d}]]$ auxiliary code wherein the physical transversal $R_Z\left(\frac{\pi}{2^{\ell+1}}\right)^\dagger$ realizes the logical transversal $R_Z\left(\frac{\pi}{2^{\ell+1}}\right)$. Given a primary CSS code $(C_1', C_2')_{\text{CSS}}$ and a target address matrix $A \in \mathbb{F}_2^{k' \times k'}$, we perform the following steps to obtain the derived code:

(1) First, we determine the number of appending matrices:
   (a) Let $1 \leq c_1 < \cdots < c_\kappa \leq k'-1$ denote the ordered indices c_i for which $\text{Supp}(A(c_i, c_i+1 : k'))$ is non-empty.
   (b) For each $i \in [\kappa]$, let $M_i = \lceil |\text{Supp}(A(c_i, c_i+1 : k'))| / \widetilde{k} \rceil$ be the number of appending matrices required for the control qubit c_i.
   (c) The total number of appending matrices is $M = \sum_{i=1}^{\kappa} M_i$. We define the corresponding matrices $G_1^{(i,j)} \in \mathbb{F}_2^{k' \times \widetilde{n}}$ and $G_2^{(i,j)} \in \mathbb{F}_2^{k_2 \times \widetilde{n}}$ for $i \in [\kappa]$ and $j \in [M_i]$, where $k_2 := \dim(\widetilde{C}_2)$.

(2) For each $i \in [\kappa]$:
   (a) Let $\mathsf{s}_i = |\text{Supp}(A(c_i, c_i+1 : k'))|$, and let $c_i < j_1 < \cdots < j_{\mathsf{s}_i} \leq k'$ denote the ordered indices in $\text{Supp}(A(c_i, c_i+1 : k'))$. Setting $j_0 := c_i$, we decompose $A(c_i, \cdot)$ into M_i

address vectors $A_{i,1}, \ldots, A_{i,M_i}$, where:

$$A_{i,p} = \left(\mathbf{0}_{j_{(p-1)\widetilde{k}}}, A\left(c_i, j_{(p-1)\widetilde{k}}+1 : j_{p\widetilde{k}}\right), \mathbf{0}_{k'-j_{p\widetilde{k}}}\right), \quad \text{for } p \in [M_i - 1] \text{ and}$$

$$A_{i,M_i} = \left(\mathbf{0}_{j_{(M_i-1)\widetilde{k}}}, A\left(c_i, j_{(M_i-1)\widetilde{k}}+1 : k'\right)\right)$$

(b) For each $j \in [M_i]$:

(i) Let $\mathsf{s}_{i,j} = |\mathrm{Supp}(A_{i,j})|$, and let $c_i < p_1 < \cdots < p_{\mathsf{s}_{i,j}} \leq k'$ denote the indices in $\mathrm{Supp}(A_{i,j})$. We define the rows $y_1^{(i,j)}, \ldots, y_{k'}^{(i,j)}$ of $\widehat{G}_1^{(i,j)}$ for each $h \in [k']$ as:

$$y_h^{(i,j)} = \begin{cases} \bigoplus_{p=1}^{\widetilde{k}} \widetilde{y}_p, & \text{if } h = c_i, \\ \widetilde{y}_q, & \text{if } h = p_q \text{ for some } q \in [\mathsf{s}_{i,j}], \\ \mathbf{0}_{\widetilde{n}}, & \text{otherwise.} \end{cases}$$

(ii) Set $\widehat{G}_2^{(i,j)} = G_{\widetilde{C}_2}$.

(iii) To cancel the residual phase induced by the addressable logical gates $\left(R_Z\left(\frac{\pi}{2^\ell}\right)_{A_1}\right)^{\widetilde{k}}$ and $R_Z\left(\frac{\pi}{2^\ell}\right)_{A_2}$, we append additional auxiliary matrices to $\widehat{G}_1^{(i,j)}$ and $\widehat{G}_2^{(i,j)}$, as summarized in Section 5.1.4. The corresponding address vectors are $A_1 = (\mathbf{0}_{c_i-1}, 1, \mathbf{0}_{k'-c_i})$ and A_2, where $A_2(q) = 1$ if and only if $q = p_r$ for $r \in \mathsf{s}_{i,j}$. This step yields the final appending matrices $G_1^{(i,j)}$ and $G_2^{(i,j)}$.

(3) The generator matrices of the final derived CSS code $(C_1, C_2)_{\mathsf{CSS}}$ are given by

$$G_{C_2} = \begin{bmatrix} G_{C_2'} & 0 & \cdots & 0 \\ 0 & G_2^{(1,1)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G_2^{(\kappa,M_\kappa)} \end{bmatrix} \quad \text{and} \quad G_{C_1/C_2} = \begin{bmatrix} G_{C_1'/C_2'} & G_1^{(1,1)} & \cdots & G_1^{(\kappa,M_\kappa)} \end{bmatrix}.$$

By Lemma 7, this procedure yields a derived CSS code with parameters $[[n' + 3\widetilde{n}M, k', \geq \min\{d', \widetilde{d}\}]]$. Within this code, the target logical gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ is implemented via the transversal physical Z-rotation

$$\prod_{i=1}^{\kappa}\prod_{j=1}^{M_i} U\left(\ell+1, w_1^{(i,j)}\right)\left(U\left(\ell+1, w_2^{(i,j)}\right)\right)^{\widetilde{k}} U\left(\ell+1, w_3^{(i,j)}\right)$$

where the vectors $w_1^{(i,j)}$, $w_2^{(i,j)}$, and $w_3^{(i,j)}$ are given by

$$w_1^{(i,j)} = \left(\mathbf{0}_{n'+3\widetilde{n}\left(\left(\sum_{p=1}^{i-1} M_p\right)+j-1\right)}, \mathbf{1}_{\widetilde{n}}, \mathbf{0}_{\widetilde{n}}, \mathbf{0}_{\widetilde{n}}, \mathbf{0}_{3\widetilde{n}\left(M_i-j+\left(\sum_{p=i+1}^{\kappa} M_p\right)\right)}\right),$$

$$w_2^{(i,j)} = \left(\mathbf{0}_{n'+3\widetilde{n}\left(\left(\sum_{p=1}^{i-1} M_p\right)+j-1\right)}, \mathbf{0}_{\widetilde{n}}, -\mathbf{1}_{\widetilde{n}}, \mathbf{0}_{\widetilde{n}}, \mathbf{0}_{3\widetilde{n}\left(M_i-j+\left(\sum_{p=i+1}^{\kappa} M_p\right)\right)}\right),$$

$$w_3^{(i,j)} = \left(\mathbf{0}_{n'+3\widetilde{n}\left(\left(\sum_{p=1}^{i-1} M_p\right)+j-1\right)}, \mathbf{0}_{\widetilde{n}}, \mathbf{0}_{\widetilde{n}}, -\mathbf{1}_{\widetilde{n}}, \mathbf{0}_{3\widetilde{n}\left(M_i-j+\left(\sum_{p=i+1}^{\kappa} M_p\right)\right)}\right).$$

As in Section 5, we conclude this section by constructing CSS codes that transversally realize any addressable logical $C\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ gate. A CSS code can realize any addressable logical $C\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ gate if and only if it can realize $C\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ between any pair of logical qubits. For the construction, we invoke the appending framework with $M = \binom{k'}{2}$, and for each pair of logical qubits $1 \leq i < j \leq k'$, define a target logical gate $U_{\mathrm{L}}^{(i,j)} = C_{i,j}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$. This amounts to instantiating the above construction with the target address matrices $A_{i,j}$ whose sole non-zero entry is $A_{i,j}(i,j) = 1$.

7. CSS Families Realizing Addressable Logical Single-Qubit Diagonal Gates

Given an arbitrary but fixed sequence of target address vectors $(A_i)_{i\in\mathbb{N}}$ with $A_i \in \mathbb{F}_2^{k_i'}$, we aim to construct a CSS family $((C_{1,i}, C_{2,i})_{\mathrm{CSS}})_{i\in\mathbb{N}}$ encoding k_i' logical qubits and transversally realizing the addressable logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_{A_i}$. To achieve this via the appending framework requires an auxiliary CSS code family in which the physical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)^\dagger$ realizes the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$. As detailed in [23], puncturing $\widehat{t}$ coordinates of a $2^{\ell+1}$-divisible classical $[\widehat{n}, \widehat{k}, \widehat{d}]$ code with dual distance $\widehat{d}^\perp$ yields such an auxiliary $[[\widehat{n} - \widehat{t}, \widehat{t}, \geq \min\{\widehat{d} - \widehat{t}, \widehat{d}^\perp - \widehat{t}\}]]$ CSS code for any integer $1 \leq \widehat{t} < \min\{\widehat{k}, \widehat{d}, \widehat{d}^\perp\}$. We construct the families for the cases $\ell = 1$ and $\ell > 1$ separately. In both cases, we assume $\widehat{t}$ is chosen to scale as $\Theta\left(\min\{\widehat{k}, \widehat{d}, \widehat{d}^\perp\}\right)$.

7.1. **Case $\ell = 1$:** For fixed constants $R, \delta \in (0,1)$, the asymptotically good family of doubly-even (4-divisible) codes from [8, 9] yields an asymptotically good CSS family $((\widetilde{C}_{1,i}, \widetilde{C}_{2,i})_{\mathrm{CSS}})_{i\in\mathbb{N}}$ with parameters $[[\widetilde{n}_i, \widetilde{k}_i \geq \widetilde{n}_i R, \widetilde{d}_i \geq \widetilde{n}_i\delta]]$, in which the physical transversal $S^\dagger$ realizes the logical transversal S. For each $i \in \mathbb{N}$, selecting $(\widetilde{C}_{1,i}, \widetilde{C}_{2,i})_{\mathrm{CSS}}$ as both the primary and auxiliary codes satisfies the bounded support condition $|\mathrm{Supp}(A_i)| \leq \widetilde{k}_i$. Consequently, the appending-matrix construction in Section 5.1.2 yields a $[[2\widetilde{n}_i, \widetilde{k}_i, \geq \widetilde{d}_i]]$ CSS code that transversally realizes S_{A_i}. This confirms the existence of an asymptotically good CSS family supporting the addressable logical gates S_{A_i}. Note, however, that this resulting CSS family is not explicit because the underlying doubly even codes in [8, 9] are not explicit.

To construct such explicit asymptotically good families, we rely on the relaxed characterization in Proposition 11 (with $\ell = 1$) to realize addressable logical S gates. This relaxed characterization establishes that, rather than requiring doubly-even codes, it suffices to use self-orthogonal codes in the puncturing construction above. Crucially, explicit asymptotically good binary self-orthogonal codes with asymptotically good duals are known to exist [69].

7.2. **Case $\ell > 1$:** For this case, we employ the classical Reed-Muller family, $C_r := \mathrm{RM}(r, (\ell+1)r+1)$ for any integer $r \in \mathbb{N}$. By Ax's theorem [70], this code is $2^{\lceil((\ell+1)r+1)/r\rceil - 1} = 2^{\ell+1}$-divisible. Furthermore, it has blocklength $\widehat{n}_r = 2^{(\ell+1)r+1}$, dimension $\widehat{k}_r = \sum_{j=0}^{r}\binom{(\ell+1)r+1}{j} \geq 2^{2r}$, minimum distance $\widehat{d}_r = 2^{\ell r+1}$, and dual distance $\widehat{d}_r^\perp = 2^{r+1}$. Puncturing $\widehat{t} = 2^r$ coordinates results in an $[[\widetilde{n}_r = 2^{(\ell+1)r+1} - 2^r, \widetilde{k}_r = 2^r, \widetilde{d}_r \geq 2^r]]$ CSS code, $(\widetilde{C}_{1,r}, \widetilde{C}_{2,r})_{\mathrm{CSS}}$, in which the physical transversal

$R_Z\left(\frac{\pi}{2^\ell}\right)^\dagger$ realizes the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$ [23]. These represent the best parameters currently known in the literature for such CSS codes.

Because the logical dimension of the auxiliary code $(\widetilde{C}_{1,r}, \widetilde{C}_{2,r})_{\text{CSS}}$ does not scale linearly with its physical blocklength, we instantiate the primary code with an asymptotically good family of CSS codes $((C'_{1,i}, C'_{2,i})_{\text{CSS}})_{i\in\mathbb{N}}$ from [69]. For this family, as the blocklength grows arbitrarily large, the parameters $[[n'_i, k'_i, d'_i]]$ scale linearly: $k'_i \geq Rn'_i$ and $d'_i \geq \min\{d_{\min}(C'_{1,i}/C'_{2,i}), d_{\min}((C'_{2,i})^\perp)\} \geq \delta n'_i$, for fixed constants $R, \delta \in (0,1)$.

7.2.1. *CSS code families for sub-linear target vector support size:* We first construct CSS code families that transversally realize the addressable logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_{A_i}$ for $A_i \in \mathbb{F}_2^{k'_i}$, assuming a sub-linear target support $|\text{Supp}(A_i)| \leq (n'_i)^\epsilon$ for a constant $\epsilon \in (0,1)$.

For each $i \in \mathbb{N}$, we apply the appending-matrix construction to the primary CSS code $(C'_{1,i}, C'_{2,i})_{\text{CSS}}$. Because the logical dimension already scales linearly with n'_i, we seek an index r for the auxiliary code $(\widetilde{C}_{1,r}, \widetilde{C}_{2,r})_{\text{CSS}}$ such that the physical blocklength of the derived code remains $\Theta(n'_i)$.

The appending-matrix construction for single-qubit Z-rotations from Subsection 5.1 requires $g = \lceil |\text{Supp}(A_i)|/2^r \rceil$ appending matrices. By Lemma 7, the blocklength of the derived CSS code $(C_{1,i,r}, C_{2,i,r})_{\text{CSS}}$ is given by

$$n'_i + \widetilde{n}_r g = n'_i + ((2^{(\ell+1)r+1} - 2^r)\lceil |\text{Supp}(A_i)|/2^r \rceil) \;\leq\; n'_i + 2^{\ell r+1}(n'_i)^\epsilon + 2^{(\ell+1)r+1}.$$

To ensure that the blocklength of the derived code scales linearly with n'_i, we set r as

$$r := \max\left\{1, \left\lfloor \min\left\{\frac{1-\epsilon}{\ell}, \frac{1}{\ell+1}\right\} \log\big(n'_i\big) \right\rfloor - 1\right\}.$$

For sufficiently large i, we have

$$r = \left\lfloor \min\left\{\frac{1-\epsilon}{\ell}, \frac{1}{\ell+1}\right\} \log\big(n'_i\big) \right\rfloor - 1 \;<\; \min\left\{\frac{1-\epsilon}{\ell}, \frac{1}{\ell+1}\right\} \log\big(n'_i\big)$$

This allows us to bound the block length relative to n'_i as follows:

$$2^{\ell r} < (n'_i)^{\min\{1-\epsilon, \ell/(\ell+1)\}} \leq (n'_i)^{(1-\epsilon)} \quad \text{and} \quad 2^{(\ell+1)r} < (n'_i)^{\min\{((\ell+1)(1-\epsilon))/\ell, 1\}} \leq n'_i$$
$$\implies n'_i + \widetilde{n}_r g < 5n'_i.$$

Thus, the derived CSS code has physical blocklength $\Theta(n'_i)$ and preserves the logical dimension of the primary code, $k'_i \geq Rn'_i$. Its minimum distance is $\min\{d_X, d_Z\}$, where $d_X \geq \delta n'_i$ and $d_Z \geq \min\{\delta n'_i, 2^r\}$. To analyze the asymptotic distance, let $\eta = \min\left\{\frac{1-\epsilon}{\ell}, \frac{1}{\ell+1}\right\} \in (0,1)$. Since $\eta \leq 1/(\ell+1)$, the term 2^r grows sub-linearly with respect to n'_i. Therefore, for large i, the Z-distance of the derived CSS code evaluates to

$$\min\{n'_i\delta, 2^r\} = 2^{\lfloor \eta \log(n'_i)\rfloor - 1} \geq \frac{1}{4}(n'_i)^\eta.$$

This establishes that the minimum distance of the derived code scales sub-linearly as $\Omega((n'_i)^\eta)$.

In conclusion, we constructed an explicit CSS code family with parameters $[[\Theta(n_i'), k_i' \geq n_i' R, \Omega((n_i')^\eta)]]$ that transversally realizes the addressable logical gates $R_Z\left(\frac{\pi}{2^\ell}\right)_{A_i}$. The inverse relationship between η and ϵ exposes a tradeoff: a larger target support (increasing ϵ) decreases the minimum distance, while a restricted support increases it.

7.2.2. *CSS code families for arbitrary target vector support size:* A limitation of the CSS code family constructed above is its inability to realize the logical gate $R_Z\left(\frac{\pi}{2^\ell}\right)_{A_i}$ for linearly scaling target support sizes. In particular, the special case of realizing the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$ exposes the primary bottleneck of our appending construction: the best known CSS codes realizing the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$ via transversal gates achieve only sub-linear parameters $[[\widetilde{n}_r, \Omega(\widetilde{n}_r^{1/(\ell+1)}), \Omega(\widetilde{n}_r^{1/(\ell+1)})]]$. Indeed, the existence of asymptotically good CSS codes transversally realizing the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$ remains a prominent open problem. As demonstrated for the case $\ell = 1$, resolving this open problem for larger ℓ would allow our framework to yield an asymptotically good CSS family for arbitrary support sizes. Solving the following classical coding problem, initially posed in [23] for $\ell = 3$ (8-divisible codes, see [71]), would directly resolve this quantum open problem.

Open Problem 1. *For any fixed $\ell \geq 3$, does there exist a family of asymptotically good 2^ℓ-divisible codes whose dual codes are also asymptotically good?*

7.3. **CSS families realizing any addressable logical $R_Z\left(\frac{\pi}{2^\ell}\right)$ gate.** To realize any addressable logical $R_Z\left(\frac{\pi}{2^\ell}\right)$ gate, we construct auxiliary matrices to apply $R_Z\left(\frac{\pi}{2^\ell}\right)$ to each logical qubit independently and append them to the primary code. This again requires auxiliary codes wherein the physical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)^\dagger$ realizes the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$. Because each appending-matrix targets only a single logical qubit, we set $\widehat{t} = 1$. As before, we consider the cases $\ell = 1$ and $\ell > 1$ separately.

7.3.1. *Case $\ell = 1$:* We use the asymptotically good primary CSS family from Section 7.1, which has parameters $[[\widetilde{n}_i, \Omega(\widetilde{n}_i), \Omega(\widetilde{n}_i)]]$, and realizes the logical transversal S via physical transversal $S^\dagger$. For the auxiliary codes (with $\widehat{t} = 1$), we utilize a CSS family with parameters $[[\Theta(\widetilde{n}_i), 1, \Omega(\widetilde{n}_i)]]$ in which the physical transversal $S^\dagger$ realizes the logical S. Applying the appending-matrix construction to realize S independently on each logical qubit yields a CSS family with parameters $[[\Theta((\widetilde{n}_i)^2), \Omega(\widetilde{n}_i), \Omega(\widetilde{n}_i)]]$. Rescaling the parameters in terms of the total physical blocklength, we obtain a final CSS family with parameters $[[\Theta(\widetilde{n}_i), \Omega((\widetilde{n}_i)^{1/2}), \Omega((\widetilde{n}_i)^{1/2})]]$ that can realize any addressable logical S gate. While this resulting family is not explicit, we can construct such explicit families achieving the same asymptotic parameters by using explicit asymptotically good self-orthogonal codes with asymptotically good duals, as discussed in Section 7.1.

7.3.2. *Case $\ell > 1$:* From Section 7.2, we use the asymptotically good primary CSS family with parameters $[[n_i', k_i' \geq n_i' R, d_i' \geq n_i' \delta]]$. For the auxiliary family (with $\widehat{t} = 1$), we consider the

Reed-Muller based CSS family with parameters $[[2^{(\ell+1)r+1}-1, 1, 2^{r+1}-1]]$ in which the physical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)^\dagger$ realizes logical $R_Z\left(\frac{\pi}{2^\ell}\right)$. Performing the appending construction with these codes, Lemma 7 guarantees a CSS code family with parameters $[[n_i' + (2^{(\ell+1)r+1}-1)k_i', k_i', \geq \min(d_i', 2^{r+1}-1)]]$ that realizes any addressable logical $R_Z\left(\frac{\pi}{2^\ell}\right)$ gate.

Because k_i' grows linearly with n_i', any choice of r where $2^{(\ell+1)r}$ grows with n_i' results in a superlinear overall physical blocklength. To balance this scaling against the minimum distance, we set

$$r = \max\left\{1, \lfloor(\epsilon/(\ell+1))\log\left(n_i'\right)\rfloor\right\},$$

for a constant $\epsilon > 0$. This choice bifurcates the asymptotic scaling into two regimes depending on which code limits the overall minimum distance:

(1) $\epsilon \in (0, \ell+1)$. The auxiliary code limits the distance, yielding parameters $[[\Theta((n_i')^{1+\epsilon}), \Omega(n_i'), \Omega((n_i')^{\epsilon/(\ell+1)})]]$.

(2) $\epsilon \geq \ell+1$. The primary code limits the distance, yielding parameters $[[\Theta((n_i')^{1+\epsilon}), \Omega(n_i'), \Omega(n_i')]]$.

Rescaling these families to a linear physical blocklength $\Theta(n_i')$ yields a family of CSS codes with parameters $[[\Theta(n_i'), \Omega((n_i')^{1/(1+\epsilon)}), \Omega((n_i')^{\epsilon/((\ell+1)(1+\epsilon))})]]$ for $\epsilon \in (0, \ell+1)$, and $[[\Theta(n_i'), \Omega((n_i')^{1/(1+\epsilon)}), \Omega((n_i')^{1/(1+\epsilon)})]]$ for $\epsilon \geq \ell+1$, realizing any addressable logical $R_Z\left(\frac{\pi}{2^\ell}\right)$ gate.

Finally, similar to the $\ell = 1$ case, an affirmative resolution to Open Problem 1 yields a CSS code family with parameters $[[\Theta(n_i'), \Omega((n_i')^{1/2}), \Omega((n_i')^{1/2})]]$ that transversally realizes any addressable logical $R_Z\left(\frac{\pi}{2^\ell}\right)$ gate. Note that for multi-controlled-Z rotations, analogous CSS families can be constructed for both a fixed target sequence of tensors and any arbitrary addressable gate.

We remark that the CSS families constructed in this section are not low-density parity-check (LDPC), as the weights of all non-trivial stabilizers in CSS codes derived from punctured classical divisible codes scale with the blocklength. If both the primary and auxiliary codes possess LDPC X-stabilizer generators, our construction yields a derived CSS family with an LDPC X-stabilizer check matrix. However, because we guarantee the Z-distance by bounding the minimum weight of the entire logical-Z space (which includes all non-trivial Z-stabilizers), we must sacrifice the LDPC property for the Z-stabilizer check matrix. Indeed, constructing LDPC auxiliary codes that transversally realize the logical transversal $R_Z\left(\frac{\pi}{2^\ell}\right)$ is an open problem.

8. Conclusion

To summarize, in this paper, we characterized CSS codes that realize target logical diagonal gates via transversal physical Z-rotations. We then introduced the appending framework, which extends a primary CSS code using appending matrices to transversally realize an arbitrary number of such target gates. Because CSS codes can only realize logical single-qubit Z-rotations and multi-controlled-Z rotations, we detailed appending-matrix constructions for both these addressable

cases. Finally, we constructed CSS code families that realize addressable logical single-qubit Z-rotations.

We lay out some directions for future work here. As noted in Section 1, the construction of CSS codes that transversally realize the full logical Clifford group relies on an inefficient combination of ℓ independent code blocks, where each block individually realizes this group. An important direction for future work then is to construct single-block codes with shorter physical blocklengths that achieve this. Moreover, improving the asymptotic parameters of such families remains an open problem—specifically, determining whether there exist such families whose rate or minimum distance scales better than the square root of the physical blocklength.

While the appending framework can be utilized to flexibly realize any desired set of logical gates, the resulting blocklengths are generally not minimal because each appended support is exclusively dedicated to a specific target gate. To achieve shorter blocklengths, one could bypass appending matrices and directly construct CSS codes that satisfy the characterization conditions, either analytically or via computational search. However, satisfying these simultaneous constraints for multiple target gates within a tight blocklength is inherently difficult. Consequently, length-optimized codes typically sacrifice flexibility and only support a restricted set of target gates.

To maintain a robust Z-distance in the appending framework, the minimum weight of the entire logical-Z space, including all non-trivial Z-stabilizers, must remain high. While this requirement is acceptable for the shorter blocklengths used in practical quantum computing, it constitutes the primary obstacle to constructing LDPC CSS families that realize target logical diagonal gates. Therefore, a direction for future research is developing alternative constructions that circumvent the need to bound the weight of all logical-Z operators.

An affirmative resolution to Open Problem 1 would yield improved asymptotic parameters for all CSS families derived from Reed-Muller based codes in this work (the case $\ell > 1$ for single-qubit Z-rotations).

Finally, no existing CSS codes in the literature utilize the relaxed characterizations derived in Sections 5.2 and A.2. Exploiting these conditions could yield improved codes for realizing target logical diagonal gates. Currently, we can only leverage these relaxations for S and CZ gates, where punctured self-orthogonal codes can directly replace punctured doubly-even codes.

Appendix A. Addressable Logical Multi-Controlled-Z Rotations in CSS Codes

For integers $m \geq 2$ and $\ell \geq 0$, we provide an alternative characterization of CSS codes that transversally support the addressable logical gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$. This characterization motivates our appending-matrix construction for multi-controlled-Z rotations. While Proposition 6 permits any $p \geq \ell + m - 1$, we restrict our focus to $p = \ell + m - 1$.

Analogous to Lemma 9, the following technical lemma reduces modular constraints on the entire code to equivalent conditions on its basis vectors.

Lemma 13. *Let $C \subseteq \mathbb{F}_2^n$ be a binary code with basis $\{x_1, \ldots, x_k\}$, and let $A \in \mathbb{F}_2^{k\times\cdots\times k}$ be a fixed order-m tensor. For any $a \in \mathbb{F}_2^k$, define the corresponding codeword $x_a = \bigoplus_{i=1}^k a(i)x_i$. For any integer vector $w \in \mathbb{Z}^n$, the modular condition*

$$w \cdot x_a = 2^{m-1} \sum_{1\le i_1<\cdots<i_m\le k} A(i_1, \ldots, i_m)a(i_1)\cdots a(i_m) \pmod{2^{\ell+m}} \tag{24}$$

holds for all $a \in \mathbb{F}_2^k$ if and only if for all ordered indices $1 \le i_1 < \cdots < i_j \le k$ with $j \in [\ell + m]$, the following modular equation is satisfied:

$$w \cdot \left(x_{i_1} * \cdots * x_{i_j}\right) = \begin{cases} (-1)^{m-1}A(i_1, \ldots, i_m) \pmod{2^{\ell+1}}, & \text{if } j = m, \\ 0 \pmod{2^{\ell+m-j+1}}, & \text{otherwise.} \end{cases} \tag{25}$$

The proof of this lemma closely follows that of Lemma 9 and is provided in Subsection A.3.

Applying the Lemma to Proposition 6 yields the following proposition.

Proposition 14. *Let $(C_1, C_2)_{\mathsf{CSS}}$ be an $[[n, k]]$ CSS code, where $\{y_1, \ldots, y_k\}$ is a basis of C_1/C_2. For any order-m target addressability tensor $A \in \mathbb{F}_2^{k\times\cdots\times k}$, the code realizes the addressable logical gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ via a transversal physical Z-rotation $U(\ell + m - 1, w)$ if and only if for all $x \in C_2$, $a \in \mathbb{F}_2^k$, and ordered indices $1 \le i_1 < \cdots < i_j \le k$ with $j \in [\ell+m]$, the following modular equations hold:*

$$w \cdot x = 0 \pmod{2^{\ell+m}}, \qquad w \cdot (x * y_a) = 0 \pmod{2^{\ell+m-1}},$$

$$w \cdot \left(y_{i_1} * \cdots * y_{i_j}\right) = \begin{cases} (-1)^{m-1}A(i_1, \ldots, i_m) \pmod{2^{\ell+1}}, & \text{if } j = m, \\ 0 \pmod{2^{\ell+m-j+1}}, & \text{otherwise.} \end{cases}$$

A.1. **Appending Matrices for a Targeted Addressable Logical Gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$.** Our construction of appending matrices for logical $(m-1)$-controlled Z-rotations proceeds inductively. We assume appending matrices have already been derived for all logical single-qubit Z-rotations and $(\widetilde{m}-1)$-controlled Z-rotations, where $\widetilde{m} < m$. The single-qubit constructions established in Subsection 5.1 serve as the base case.

Given a primary $[[n', k']]$ CSS code $(C_1', C_2')_{\mathsf{CSS}}$ and an order-m target address tensor $A \in \mathbb{F}_2^{k'\times\cdots\times k'}$, we employ the appending framework from Section 4 to construct an extended code that transversally realizes the addressable logical gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$. We construct the required appending matrices using the auxiliary $[[\widetilde{n}, \widetilde{k}, \widetilde{d}]]$ CSS code $(\widetilde{C}_1, \widetilde{C}_2)_{\mathsf{CSS}}$ from Section 5.1, in which the physical transversal $R_Z\left(\frac{\pi}{2^{\ell+m-1}}\right)^\dagger$ realizes the logical transversal $R_Z\left(\frac{\pi}{2^{\ell+m-1}}\right)$, and $\widetilde{d}_Z \ge d_{\min}(\widetilde{C}_2^\perp) \ge \widetilde{d}$.

Recall from Section 5.1 that if $\widetilde{y}_1, \ldots, \widetilde{y}_{\widetilde{k}}$ are the rows of the coset generator matrix $G_{\widetilde{C}_1/\widetilde{C}_2}$, the following hold for all $\widetilde{x} \in \widetilde{C}_2$, $a \in \mathbb{F}_2^{\widetilde{k}}$, $i \in [\widetilde{k}]$, and ordered indices $1 \le i_1 < \cdots < i_j \le \widetilde{k}$ with $j \in \{2, \ldots, \ell + m\}$:

$$w_H(\widetilde{x}) = 0 \pmod{2^{\ell+m}}, \qquad w_H(\widetilde{x} * \widetilde{y}_a) = 0 \pmod{2^{\ell+m-1}},$$

$$w_H(\widetilde{y}_i) = -1 \pmod{2^{\ell+m}}, \qquad w_H(\widetilde{y}_{i_1} * \cdots * \widetilde{y}_{i_j}) = 0 \pmod{2^{\ell+m-j+1}}. \tag{26}$$

The following proposition, which is crucial for our subsequent construction, analyzes the Schur product of $\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i$ with the rows of the coset generator matrix $G_{\widetilde{C}_1/\widetilde{C}_2}$.

Proposition 15. *Let $\widetilde{y}_1, \ldots, \widetilde{y}_{\widetilde{k}}$ denote the rows of $G_{\widetilde{C}_1/\widetilde{C}_2}$. The following modular conditions hold:*

(1)
$$w_H\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) = -\widetilde{k} \pmod{2^{\ell+m}}. \tag{27}$$

(2) *For any index $p \in [\widetilde{k}]$,*
$$w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) * \widetilde{y}_p\right) = -1 \pmod{2^{\ell+m-1}}. \tag{28}$$

(3) *For any ordered sequence of indices $1 \le p_1 < \cdots < p_q \le \widetilde{k}$ with $q \ge 2$,*
$$w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) * \left(\widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}\right)\right) = 0 \pmod{2^{\ell+m-q}}. \tag{29}$$

The proof of this proposition is provided in Subsection A.4.

A.1.1. *Reduction to a simpler special case.* We first decompose the target gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ by lexicographically ordering its control qubit tuples. Let $(1, 2, \ldots, m-1) \le \left(c_1^{(1)}, \ldots, c_{m-1}^{(1)}\right) < \cdots < \left(c_1^{(\kappa)}, \ldots, c_{m-1}^{(\kappa)}\right) \le (k'-m+2, \ldots, k')$ denote all the active control qubit tuples—that is, tuples $(i_1, \ldots, i_{m-1})$ for which there exists at least one target qubit $t \in \{i_{m-1}+1, \ldots, k'\}$ such that $A(i_1, \ldots, i_{m-1}, t) = 1$. We factor $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ into κ addressed gates $C_{A_1}^{(m-1)}(R_Z\left(\frac{\pi}{2^\ell}\right)), \ldots, C_{A_\kappa}^{(m-1)}(R_Z\left(\frac{\pi}{2^\ell}\right))$. Each tensor A_j isolates the interactions controlled by the qubit tuple $\left(c_1^{(j)}, \ldots, c_{m-1}^{(j)}\right)$ by inheriting the following tuple slice of A and vanishing elsewhere:

$$A_j(i_1, \ldots, i_{m-1}, i_m) := \begin{cases} A\left(c_1^{(j)}, \ldots, c_{m-1}^{(j)}, i_m\right), & \text{if } (i_1, \ldots, i_{m-1}) = (c_1^{(j)}, \ldots, c_{m-1}^{(j)}) \\ & \quad \text{and } i_m \in \{c_{m-1}^{(j)}+1, \ldots, k'\}, \\ 0, & \text{otherwise.} \end{cases}$$

If the appending-matrix construction is established for each factor $C_{A_j}^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$, the appending framework guarantees that each gate $C_{A_j}^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ is realized by some transversal Z-rotation $U(\ell+m-1, w_j)$ in the final extended code. The composition $\prod_{j=1}^{\kappa} U(\ell+m-1, w_j)$ then realizes the full target gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$. Consequently, without loss of generality, we assume that the only active control qubit tuple is $(1, \ldots, m-1)$—meaning, the logical $C^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$

gates act exclusively with the first qubit tuple $(1, 2, \ldots, m-1)$ as the control qubits and a subset of the remaining qubits as targets.

Furthermore, we restrict the construction to target matrices satisfying $|\mathrm{Supp}(A(1, \ldots, m-1, m : k'))| \leq \widetilde{k}$, where

$$A(1, \ldots, m-1, m : k') := (A(1, \ldots, m-1, m), A(1, \ldots, m-1, m+1), \ldots, A(1, \ldots, m-1, k')).$$

If the target support exceeds $\widetilde{k}$, the flexibility of the appending framework allows us to partition the addressable gate $C_A\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ into $g = \lceil |\mathrm{Supp}(A(1, \ldots, m-1, m : k'))| / \widetilde{k} \rceil$ factors, each acting from the control qubit tuple $(1, \ldots, m-1)$ onto at most $\widetilde{k}$ targets in $\{m, \ldots, k'\}$. Constructing appending matrices independently for each factor and composing their corresponding physical Z-rotations realizes the target gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ within the final extended code.

A.1.2. *The appending-matrix construction in the special case of* $|\mathrm{Supp}(A(1, \ldots, m-1, m : k'))| \leq \widetilde{k}$. To simplify the presentation of the construction, we further assume that $A(i_1, \ldots, i_{m-1}, i_m) = 1$ if and only if $(i_1, \ldots, i_{m-1}) = (1, \ldots, m-1)$ and $i_m \in \{m, m+1, \ldots, \mathsf{s}+m-1\}$, where $\mathsf{s} := |\mathrm{Supp}(A(1, \ldots, m-1, m : k'))| \leq \widetilde{k}$. This effectively designates $\{m, \ldots, \mathsf{s}+m-1\}$ as the set of all target qubits. Again, recall that we are given a primary $[[n', k']]$ CSS code $(C_1', C_2')_{\mathsf{CSS}}$, and that we will obtain appending matrices G_1 and G_2 using an auxiliary $[[\widetilde{n}, \widetilde{k}, \widetilde{d}]]$ CSS code $(\widetilde{C}_1, \widetilde{C}_2)_{\mathsf{CSS}}$ for which the modular equations in (26) hold. We define the rows $y_1, \ldots, y_{k'}$ of G_1 as follows: we set $y_j = \bigoplus_{p=1}^{\widetilde{k}} \widetilde{y}_p$ for the control qubit rows $j \in [m-1]$, assign $y_j = \widetilde{y}_{j-m+1}$ for target qubits $m \leq j \leq \mathsf{s}+m-1$, and set $y_j = 0$ for all remaining rows $\mathsf{s}+m-1 < j \leq k'$.

Next, we set $G_2 = G_{\widetilde{C}_2}$, and let $(\widehat{C}_1, \widehat{C}_2)_{\mathsf{CSS}}$ denote the CSS code obtained by applying the appending-matrix construction to $(C_1', C_2')_{\mathsf{CSS}}$.

A.1.3. $U(\ell+m-1, w)$ *is a logical operator for* $(\widehat{C}_1, \widehat{C}_2)_{\mathsf{CSS}}$. Consider the transversal Z-rotation $U(\ell+m-1, w)$ with $w = (\mathbf{0}_{n'}, (-1)^m \mathbf{1}_{\widetilde{n}})$. By construction, $G_2 = G_{\widetilde{C}_2}$ and the rows of G_1 are drawn from $\widetilde{C}_1/\widetilde{C}_2$. Consequently, for any $x = (x', \widetilde{x}) \in C_2$ and $y_a = (y', \widetilde{y}) \in C_1/C_2$ (with $a \in \mathbb{F}_2^{k'}$), we have $\widetilde{x} \in \widetilde{C}_2$ and $\widetilde{y} \in \widetilde{C}_1/\widetilde{C}_2$. By (26), this guarantees

$$w \cdot x = 0 \pmod{2^{\ell+m}} \quad \text{and} \quad w \cdot (x * y_a) = 0 \pmod{2^{\ell+m-1}}.$$

Remark 4 then establishes $U(\ell+m-1, w)$ as a logical operator on $(\widehat{C}_1, \widehat{C}_2)_{\mathsf{CSS}}$.

A.1.4. $U(\ell+m-1, w)$ *realizes* $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ *up to a logical phase.* For any logical state $|a\rangle$ with $a \in \mathbb{F}_2^{k'}$, the action of $U(\ell+m-1, w)$ is given by

$$U(\ell+m-1, w)\,|a\rangle_L = \exp\left\{ \iota \frac{\pi}{2^{\ell+m-1}} \left(w \cdot \left(\bigoplus_{i=1}^{k'} a(i) y_i \right) \right) \right\} |a\rangle_L$$

$$= \exp\left\{ \iota \frac{\pi}{2^{\ell+m-1}} \left(\sum_{i=1}^{k'} (-1)^{i+m-1} 2^{i-1} \left(\sum_{1 \leq j_1 < \cdots < j_i \leq k'} \left(\prod_{p=1}^{i} a(j_p) \right) w_H\left(y_{j_1} * \cdots * y_{j_i}\right) \right) \right) \right\} |a\rangle_L .$$

We next show that the t-fold Schur products for $t > m$ vanish in the phase exponent, while the m-fold products exactly realize the target gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$. The t-fold products for $t < m$, however, introduce an extraneous logical phase, which we will eliminate in the next subsection.

To see why the t-fold Schur products for $t > m$ vanish, first consider any ordered sequence of $t > m$ target qubit indices $m \leq p_1 < \cdots < p_t \leq \mathsf{s} + m - 1$. Condition (26) implies

$$w_H(y_{p_1} * \cdots * y_{p_t}) = w_H(\widetilde{y}_{p_1-m+1} * \cdots * \widetilde{y}_{p_t-m+1}) = 0 \pmod{2^{\ell+m-t+1}}.$$

On the other hand, if the Schur product combines the control qubit indices $1 \leq p_1 < \cdots < p_q \leq m-1$ with $q \geq 1$ and target qubit indices $m \leq r_1 < \cdots < r_t \leq \mathsf{s} + m - 1$ with $t \geq m$ (and $q + t > m$), (29) yields

$$\begin{aligned} w_H(y_{p_1} * \cdots * y_{p_q} * y_{r_1} * \cdots * y_{r_t}) &= w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) * \widetilde{y}_{r_1-m+1} * \cdots * \widetilde{y}_{r_t-m+1}\right) \\ &= 0 \pmod{2^{\ell+m-t}} \\ &= 0 \pmod{2^{\ell+m-q-t+1}}. \end{aligned}$$

Because only the first $\mathsf{s}+m-1$ rows of G_1 are non-zero, the preceding modular equations guarantee that all t-fold Schur products for $t > m$ vanish in the phase exponent. Consequently, the action of the physical rotation $U(\ell + m - 1, w)$ reduces to

$$\begin{aligned} &U(\ell + m - 1, w)\,|a\rangle_L \\ &\quad = \exp\left\{\iota \frac{\pi}{2^{\ell+m-1}} \left(\sum_{i=1}^{m} (-1)^{i+m-1} 2^{i-1} \left(\sum_{1 \leq j_1 < \cdots < j_i \leq k'} \left(\prod_{p=1}^{i} a(j_p)\right) w_H\left(y_{j_1} * \cdots * y_{j_i}\right)\right)\right)\right\} |a\rangle_L . \end{aligned}$$

We now evaluate the m-fold Schur products of non-zero rows of G_1 in the phase exponent. For any control qubit indices $1 \leq p_1 < \cdots < p_q \leq m-1$ and target qubit indices $m \leq r_1 < \cdots < r_t \leq \mathsf{s} + m - 1$ with $q + t = m$. We distinguish three cases: $q = 0$, $1 \leq q < m-1$ and $q = m-1$. If $q = 0$, then by (26), these m-fold Schur products vanish:

$$w_H(y_{r_1} * \cdots * y_{r_t}) = w_H(\widetilde{y}_{r_1-1} * \cdots * \widetilde{y}_{r_t-m+1}) = 0 \pmod{2^{\ell+m-r+1}}.$$

If $q < m-1$, then $t \geq 2$. Consequently, by (29), these products similarly vanish:

$$\begin{aligned} w_H(y_{p_1} * \cdots * y_{p_q} * y_{r_1} * \cdots * y_{r_t}) &= w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) * \widetilde{y}_{r_1-m+1} * \cdots * \widetilde{y}_{r_t-m+1}\right) \\ &= 0 \pmod{2^{\ell+m-t}} \\ &= 0 \pmod{2^{\ell+1}}. \end{aligned}$$

If $q = m-1$, all control qubits are included, and (28) yields

$$w_H(y_{p_1} * \cdots * y_{p_{m-1}} * y_{r_1}) = w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) * \widetilde{y}_{r_1-m+1}\right) = -1 \pmod{2^{\ell+m-1}} = -1 \pmod{2^{\ell+1}}.$$

Substituting these evaluations into the m-fold product terms, and recalling that $A(i_1,\ldots,i_{m-1},i_m) = 1$ if and only if $(i_1,\ldots,i_{m-1}) = (1,\ldots,m-1)$ and $i_m \in \{m, m+1, \ldots, \mathsf{s}+m-1\}$, we see that the logical action of the transversal physical rotation $U(\ell+m-1, w)$ further reduces to

$$\begin{aligned}
&U(\ell+m-1,w)\,|a\rangle_L \\
&= \exp\Bigg\{\iota\frac{\pi}{2^{\ell+m-1}}\Bigg(\Bigg(\sum_{i=1}^{m-1}(-1)^{i+m-1}2^{i-1}\Bigg(\sum_{1\le j_1<\cdots<j_i\le k'}\Bigg(\prod_{p=1}^{i}a(j_p)\Bigg)w_H\left(y_{j_1}*\cdots*y_{j_i}\right)\Bigg)\Bigg)+ \\
&\qquad\qquad\Bigg(2^{m-1}\sum_{1\le j_1<\cdots<j_m\le k'}A(j_1,\ldots,j_m)a(j_1)\cdots a(j_m)\Bigg)\Bigg)\Bigg\}|a\rangle_L \\
&= \exp\Bigg\{\iota\frac{\pi}{2^{\ell+m-1}}\Bigg(\sum_{i=1}^{m-1}(-1)^{i+m-1}2^{i-1}\Bigg(\sum_{1\le j_1<\cdots<j_i\le k'}\Bigg(\prod_{p=1}^{i}a(j_p)\Bigg)w_H\left(y_{j_1}*\cdots*y_{j_i}\right)\Bigg)\Bigg)\Bigg\} \\
&\qquad\qquad C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)|a\rangle_L\,.
\end{aligned}$$

Thus, we have successfully realized the target gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$ up to a logical phase.

A.1.5. *Eliminating the residual logical phase.* The residual logical phase arises entirely from the $\widetilde{m}$-fold Schur products for $2 \le \widetilde{m} < m$, which correspond to addressable logical $(\widetilde{m}-1)$-controlled-Z rotations, and the linear terms, which correspond to addressable logical single-qubit Z-rotations. Let U_{L}' denote the composite logical gate induced by this extraneous phase. By our inductive construction hypothesis on the number of control qubits, we can systematically construct appending matrices for both these $\widetilde{m}$-controlled-Z and single-qubit Z-rotations to transversally realize the inverse gate, $(U_{\mathrm{L}}')^\dagger$. Incorporating the appending matrices for $(U_{\mathrm{L}}')^\dagger$ cancels the residual phase, yielding the final extended CSS code that transversally realizes the target gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^\ell}\right)\right)$.

To make this cancellation concrete for the t-fold Schur products, we explicitly evaluate the residual logical phases for $t=1$ and $t=2$. For the linear terms ($t=1$), by (26) and (27), we have $w_H(\widetilde{y}_i) = -1 \pmod{2^{\ell+m}}$ and $w_H\left(\bigoplus_{i=1}^{\widetilde{k}}\widetilde{y}_i\right) = -\widetilde{k} \pmod{2^{\ell+m}}$. Consequently, the linear terms induce the addressable single-qubit Z-rotations $\left(R_Z\left(\frac{\pi}{2^{\ell+m-1}}\right)_{A_1}^{(-1)^{m-1}}\right)^{\widetilde{k}}$ and $R_Z\left(\frac{\pi}{2^{\ell+m-1}}\right)_{A_2}^{(-1)^{m-1}}$, where $A_1 = (\mathbf{1}_{m-1}, \mathbf{0}_{k'-m+1})$ and $A_2 = (\mathbf{0}_{m-1}, \mathbf{1}_{\mathsf{s}}, \mathbf{0}_{k'-\mathsf{s}-m+1})$.

We systematically eliminate the phase induced by these linear terms by appending auxiliary matrices realizing $R_Z\left(\frac{\pi}{2^{\ell+m-1}}\right)_{A_1}$ and $R_Z\left(\frac{\pi}{2^{\ell+m-1}}\right)_{A_2}$. Depending on the parity of m, we may

specifically need to realize the inverses of this gate; crucially, Remark 3 guarantees that this inversion is straightforward: if $U(p, w)$ realizes a logical gate U_{L}, then $U(p, -w)$ realizes $U_{\mathrm{L}}^{\dagger}$.

Similarly, evaluating the 2-fold product terms using $w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) * \widetilde{y}_j\right) = -1 \pmod{2^{\ell+m-1}}$ and $w_H\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) = -\widetilde{k} \pmod{2^{\ell+m-1}}$ identifies the residual addressable 1-controlled-Z rotations as $\left(C_{A_1}\left(R_Z\left(\frac{\pi}{2^{\ell+m-2}}\right)\right)^{(-1)^m}\right)^{\widetilde{k}}$ and $C_{A_2}\left(R_Z\left(\frac{\pi}{2^{\ell+m-2}}\right)\right)^{(-1)^m}$, where the non-zero entries of A_1 and A_2 in $\mathbb{F}_2^{k' \times k'}$ are

$$A_1(i,j) = 1 \text{ for } 1 \le i < j \le m-1 \quad \text{and} \quad A_2(i,j) = 1 \text{ for } i \in [m-1],\ m \le j \le \mathsf{s}+m-1.$$

Appending auxiliary matrices that realize the inverse of these gates cancels the phase induced by the 2-fold products. Iterating this cancellation procedure through all remaining t-fold products ($3 \le t < m$) completely eliminates the extraneous logical phases. This completes our construction, yielding the final extended CSS code $(C_1, C_2)_{\mathsf{CSS}}$ that transversally realizes the target gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^{\ell}}\right)\right)$.

As in Section 6, we can construct CSS codes that transversally realize any addressable logical $C^{(m-1)}\left(R_Z\left(\frac{\pi}{2^{\ell}}\right)\right)$ gate. A CSS code can realize any addressable logical $C^{(m-1)}\left(R_Z\left(\frac{\pi}{2^{\ell}}\right)\right)$ gate if and only if it can realize $C^{(m-1)}\left(R_Z\left(\frac{\pi}{2^{\ell}}\right)\right)$ across any logical qubit tuple $(i_1, \ldots, i_m)$, where $1 \le i_1 < \cdots < i_m \le k'$. For the construction, we invoke the appending framework with $M = \binom{k'}{m}$, and for every qubit tuple, define a target logical gate $U_{\mathrm{L}}^{(i_1,\ldots,i_m)} = C_{i_1,\ldots,i_m}\left(R_Z\left(\frac{\pi}{2^{\ell}}\right)\right)$. This amounts to instantiating the above construction with the target address tensors $A_{i_1,\ldots,i_m}$ whose sole non-zero entry is $A_{i_1,\ldots,i_m}(i_1, \ldots, i_m) = 1$.

A.2. **Relaxed Modular Equations to Realize Addressable Logical Multi-Controlled-Z Rotations.** As in Section 5.2, we demonstrate that the system of modular equations in Proposition 14 can be relaxed. Applying Lemma 17 in Appendix B to Proposition 14 yields the following alternative characterization:

Proposition 16. *Let $(C_1, C_2)_{\mathsf{CSS}}$ be an $[[n, k]]$ CSS code, where $\{x_1, \ldots, x_{k_2}\}$ and $\{y_1, \ldots, y_k\}$ are bases for C_2 and C_1/C_2, respectively. For any order-m target address tensor $A \in \mathbb{F}_2^{k \times \cdots \times k}$, the code realizes the addressable logical gate $C_A^{(m-1)}\left(R_Z\left(\frac{\pi}{2^{\ell}}\right)\right)$ via a transversal physical Z-rotation $U(\ell+m-1, w')$ for some $w' \in \mathbb{Z}^n$ if and only if there exists a vector $w \in \mathbb{Z}^n$ such that for all $a \in \mathbb{F}_2^k$, $p \in [k_2]$, ordered indices $1 \le p_1 < \cdots < p_q \le k_2$ with $q \in \{2, \ldots, k_2\}$, $i \in [k]$, ordered indices $1 \le i_1 < \cdots < i_j \le k$, the following conditions hold:*

$$w \cdot x_p = 0 \pmod{2^{\ell+m-1}}, \qquad w \cdot (x_{p_1} * \cdots * x_{p_q}) = 0 \pmod{2^{\ell+m-q+1}},$$
$$w \cdot (x * y_a) = 0 \pmod{2^{\ell+m-1}},$$

$$w \cdot \left(y_{i_1} * \cdots * y_{i_j}\right) = \begin{cases} (-1)^{m-1} A(i_1, \ldots, i_m) \pmod{2^{\ell+1}}, & \text{if } j = m, \\ 0 \pmod{2^{\ell+m-1}}, & \text{if } j = 1, \\ 0 \pmod{2^{\ell+m-j+1}}, & \text{otherwise.} \end{cases}$$

A.3. **Proof of Lemma 13.**

Proof. We establish necessity via a two-phase induction on $j \in [\ell + m]$: the first phase covers $j = 1$ through $m - 1$ (with base case $j = 1$), while the second spans $j = m$ through $\ell + m$ (with base case $j = m$). We begin with the base case of the first phase: for any $i \in [k]$, evaluating (24) at $a = e_i$ establishes the base case

$$w \cdot x_i = 0 \pmod{2^{\ell+m}},$$

where the last equality holds because $m \geq 2$. For the induction step, assume the result holds for all p where $1 \leq p \leq j < m - 1$. Then, for any ordered indices $1 \leq i_1 < \cdots < i_p \leq k$, we have

$$w \cdot \left(x_{i_1} * \cdots * x_{i_p}\right) = 0 \pmod{2^{\ell+m-p+1}}. \tag{30}$$

Now, for any ordered indices $1 \leq i_1 < \cdots < i_{j+1} \leq k$, evaluating (24) at $a = \bigoplus_{p=1}^{j+1} e_{i_p}$ yields

$$w \cdot \left(\bigoplus_{p=1}^{j+1} x_{i_p}\right) = 0 \pmod{2^{\ell+m}}. \tag{31}$$

By expanding the binary addition on the left-hand side of (31) and applying the induction hypothesis (30), we obtain

$$w \cdot \left(\bigoplus_{p=1}^{j+1} x_{i_p}\right) = (-2)^j w \cdot \left(x_{i_1} * \cdots * x_{i_{j+1}}\right) \pmod{2^{\ell+m}}. \tag{32}$$

Equating (31) and (32), we complete the phase one induction argument:

$$w \cdot \left(x_{i_1} * \cdots * x_{i_{j+1}}\right) = 0 \pmod{2^{\ell+m-j}}. \tag{33}$$

We now turn to the base case of the second phase ($j = m$). For any ordered indices $1 \leq i_1 < \cdots < i_m \leq k$, evaluating (24) at $a = \bigoplus_{p=1}^{m} e_{i_p}$ yields

$$w \cdot \left(\bigoplus_{p=1}^{m} x_{i_p}\right) = 2^{m-1} A(i_1, \ldots, i_m) \pmod{2^{\ell+m}}. \tag{34}$$

Expanding the binary addition on the left-hand side of (34) and incorporating the modular conditions for $1 \leq j \leq m - 1$, we arrive at the base case, $j = m$:

$$w \cdot (x_{i_1} * \cdots * x_{i_m}) = (-1)^{m-1} A(i_1, \ldots, i_m) \pmod{2^{\ell+1}}. \tag{35}$$

For the second phase induction, assume the result holds for all p where $m \le p \le j < \ell + m$. Then, for any ordered indices $1 \le i_1 < \cdots < i_p \le k$, we have

$$w \cdot (x_{i_1} * \cdots * x_{i_p}) = \begin{cases} (-1)^{m-1} A(i_1, \ldots, i_m) \pmod{2^{\ell+1}}, & \text{if } p = m, \\ 0 \pmod{2^{\ell+m-p+1}}, & \text{otherwise.} \end{cases} \tag{36}$$

Now, for any ordered indices $1 \le i_1 < \cdots < i_{j+1} \le k$, evaluating (24) at $a = \bigoplus_{p=1}^{j+1} e_{i_p}$, we obtain

$$w \cdot \left(\bigoplus_{p=1}^{j+1} x_{i_p} \right) = 2^{m-1} \sum_{1 \le q_1 < \cdots < q_m \le j+1} A(i_{q_1}, \ldots, i_{q_m}) \pmod{2^{\ell+m}}. \tag{37}$$

By expanding the binary addition on the left-hand side of (37) and applying the induction hypothesis (36), as well as the first phase modular conditions, we obtain

$$w \cdot \left(x_{i_1} * \cdots * x_{i_{j+1}} \right) = 0 \pmod{2^{\ell+m-j}},$$

completing the induction and establishing necessity. To prove sufficiency, we begin by expanding the binary addition on the left-hand side of (24) into an integer sum:

$$w \cdot \left(\bigoplus_{i=1}^{k} a(i) x_i \right) = \sum_{j=1}^{k} (-2)^{j-1} \left(\sum_{1 \le i_1 < \cdots < i_j \le k} \left(\prod_{p=1}^{j} a(i_p) \right) w \cdot (x_{i_1} * \cdots * x_{i_j}) \right).$$

Because we evaluate this expression modulo $2^{\ell+m}$, any term in the outer summation with $j \ge \ell + m + 1$ vanishes. Substituting (25) into the remaining terms completes the proof:

$$w \cdot \left(\bigoplus_{i=1}^{k} a(i) x_i \right) = 2^{m-1} \sum_{1 \le i_1 < \cdots < i_m \le k} \left(\prod_{p=1}^{m} a(i_p) \right) A(i_1, \ldots, i_m) \pmod{2^{\ell+m}}.$$

□

A.4. **Proof of Proposition 15.**

Proof. By the inclusion-exclusion principle,

$$\begin{aligned} w_H \left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i \right) &= \sum_{j=1}^{\widetilde{k}} (-2)^{j-1} \left(\sum_{1 \le i_1 < \cdots < i_j \le \widetilde{k}} w_H(\widetilde{y}_{i_1} * \cdots * \widetilde{y}_{i_j}) \right) \\ &= -1 \pmod{2^{\ell+m}}, \end{aligned}$$

where the last equality follows by (26). We now turn to prove the second claim: For any $p \in [\widetilde{k}]$, expanding the left-hand side of (28) via the inclusion-exclusion principle alongside the idempotent property $\widetilde{y}_p * \widetilde{y}_p = \widetilde{y}_p$ yields

$$w_H \left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i \right) * \widetilde{y}_p \right) = w_H \left(\widetilde{y}_p \oplus \left(\bigoplus_{i \ne p} \widetilde{y}_i * \widetilde{y}_p \right) \right)$$

$$= w_H(\widetilde{y}_p) + w_H\left(\bigoplus_{i\neq p} \widetilde{y}_i * \widetilde{y}_p\right) - 2w_H\left(\widetilde{y}_p * \left(\bigoplus_{i\neq p} \widetilde{y}_i * \widetilde{y}_p\right)\right)$$
$$= w_H(\widetilde{y}_p) - w_H\left(\bigoplus_{i\neq p} \widetilde{y}_i * \widetilde{y}_p\right)$$

Expanding the final term above via the inclusion-exclusion formula, we obtain

$$w_H\left(\bigoplus_{i\neq p} \widetilde{y}_i * \widetilde{y}_p\right) = \sum_{q=1}^{\widetilde{k}-1} (-2)^{q-1} \left(\sum_{\substack{1\leq r_1<\cdots<r_q\leq\widetilde{k}:\\ r_s\neq p\ \forall s\in[q]}} w_H(\widetilde{y}_{r_1} * \cdots * \widetilde{y}_{r_q} * \widetilde{y}_p)\right)$$
$$= 0 \pmod{2^{\ell+m-1}},$$

by virtue of (26) again. Substituting this result back into our main computation gives

$$w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) * \widetilde{y}_p\right) = w_H(\widetilde{y}_p) \pmod{2^{\ell+m-1}}.$$

Since we already know from (26) that $w_H(\widetilde{y}_p) = -1 \pmod{2^{\ell+m}}$, reducing the modulus establishes

$$w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) * \widetilde{y}_p\right) = -1 \pmod{2^{\ell+m-1}},$$

completing the proof of the second claim (28).

To prove the third claim (29), consider an arbitrary ordered sequence of distinct indices $1 \leq p_1 < \cdots < p_q \leq \widetilde{k}$ with $q \geq 2$, and let $P = \{p_1, \ldots, p_q\}$. Evaluating the left-hand side of (29) based on the parity of q, we have

$$w_H\left(\left(\bigoplus_{i=1}^{\widetilde{k}} \widetilde{y}_i\right) * \left(\widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}\right)\right) = \begin{cases} w_H\left(\bigoplus_{i\in[\widetilde{k}]\setminus P} \widetilde{y}_i * \widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}\right), & \text{if } q \text{ is even,} \\ w_H\left(\left(\widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}\right) \oplus \bigoplus_{i\in[\widetilde{k}]\setminus P} \widetilde{y}_i * \widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}\right), & \text{otherwise.} \end{cases}$$

For odd q, applying inclusion-exclusion yields

$$w_H\left(\left(\widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}\right) \oplus \left(\bigoplus_{i\in[\widetilde{k}]\setminus P} \widetilde{y}_i * \widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}\right)\right)$$
$$= w_H(\widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}) - w_H\left(\bigoplus_{i\in[\widetilde{k}]\setminus P} \widetilde{y}_i * \widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}\right)$$
$$= -w_H\left(\bigoplus_{i\in[\widetilde{k}]\setminus P} \widetilde{y}_i * \widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}\right) \pmod{2^{\ell+m-q}},$$

via (26). Note that showing that the last expression above is equal to $0 \pmod{2^{\ell+m-q}}$ also resolves the case of even q. To this end, applying inclusion-exclusion once again gives us

$$w_H\left(\bigoplus_{i\in[\widetilde{k}]\setminus P} \widetilde{y}_i * \widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q}\right) = \sum_{r=1}^{\widetilde{k}-q} (-2)^{r-1} \left(\sum_{\substack{1\le s_1<\cdots<s_r\le\widetilde{k}:\\ s_t\notin P\ \forall t\in[r]}} w_H(\widetilde{y}_{s_1} * \cdots * \widetilde{y}_{s_r} * \widetilde{y}_{p_1} * \cdots * \widetilde{y}_{p_q})\right)$$
$$= 0 \pmod{2^{\ell+m-q}},$$

again by (26). This completes the proof of the third claim (29). □

APPENDIX B. LIFTING MODULAR EQUATIONS

In this appendix, we prove the lemma that provides the mechanism for obtaining Propositions 11 and 16 from Propositions 10 and 14, respectively. Specifically, the lemma demonstrates that if an integer vector w satisfies certain modulo-2^ℓ equations over a code's basis vectors, alongside modular constraints on their t-fold Schur products (for $t \ge 2$), then we can systematically refine w into a new vector w' in such a way that the basis modular equations are lifted to their modulo-$2^{\ell+1}$ counterparts while leaving unchanged the modular constraints on all the t-fold ($t \ge 2$) Schur products.

Lemma 17. *Let $C \subseteq \mathbb{F}_2^n$ be a binary code with basis $\{x_1, \ldots, x_k\}$, and let $A \in \mathbb{F}_2^k$ be a fixed vector. Suppose there exists a vector $w \in \mathbb{Z}^n$ such that for all $i \in [k]$ and ordered indices $1 \le i_1 < \cdots < i_j \le k$ with $j \in \{2, \ldots, \ell+1\}$, the following conditions hold:*

$$w \cdot x_i = A(i) \pmod{2^\ell}, \tag{38}$$

$$w \cdot (x_{i_1} * \cdots * x_{i_j}) = 0 \pmod{2^{\ell-j+2}}. \tag{39}$$

Then, there exists an integer vector $w' \in \mathbb{Z}^n$ such that

$$w' \cdot x_i = A(i) \pmod{2^{\ell+1}}, \tag{40}$$

$$w' \cdot (x_{i_1} * \cdots * x_{i_j}) = 0 \pmod{2^{\ell-j+2}}. \tag{41}$$

Proof. By Lemma 9, conditions (40) and (41) hold if and only if

$$w' \cdot x_a = w_H(a * A) \pmod{2^{\ell+1}}.$$

To proceed, we introduce a function f_1 defined over the code C as follows:

$$f_1(x_a) = w \cdot x_a \pmod{2^{\ell+1}},$$

where $x_a = \bigoplus_{p=1}^k a(p)x_p$. Expanding the dot product yields

$$w \cdot x_a = \sum_{j=1}^{k} (-2)^{j-1} \left(\sum_{1\le i_1<\cdots<i_j\le k} \left(\prod_{p=1}^{j} a(i_p)\right) w \cdot \left(x_{i_1} * \cdots * x_{i_j}\right)\right). \tag{42}$$

From (39), $f_1(x_a)$ simplifies to

$$f_1(x_a) = \sum_{i=1}^{k} a(i) w \cdot x_i \pmod{2^{\ell+1}}.$$

Furthermore, (38) requires that for each $i \in [k]$, the difference $w \cdot x_i - A(i)$ must be either 0 or $2^\ell \pmod{2^{\ell+1}}$. Factoring 2^ℓ from these terms, the expression evaluates to

$$\begin{aligned} f_1(x_a) &= \left(w_H(a * A) \pmod{2^{\ell+1}} \right) + \left(\sum_{i=1}^{k} a(i) \left(w \cdot x_i - A(i) \right) \pmod{2^{\ell+1}} \right) \\ &= \left(w_H(a * A) \pmod{2^{\ell+1}} \right) + \left(2^\ell \left(\sum_{i=1}^{k} a(i) \frac{(w \cdot x_i - A(i))}{2^\ell} \pmod 2 \right) \pmod{2^{\ell+1}} \right). \end{aligned} \tag{43}$$

Next, we isolate the modulo-2 summation by defining the function f_2 as

$$f_2 \left(\bigoplus_{i=1}^{k} a(i) x_i \right) = \sum_{i=1}^{k} a(i) \frac{w \cdot x_i - A(i)}{2^\ell} \pmod 2.$$

As f_2 is a linear transformation from C to $\mathbb{F}_2$, there exists a vector $w_1 \in \mathbb{F}_2^n$ such that

$$f_2(x_a) = w_1 \cdot x_a \pmod 2.$$

Substituting this dot product representation back into (43) yields

$$(w - 2^\ell w_1) \cdot x_a = w_H(a * A) \pmod{2^{\ell+1}}.$$

Setting $w' = w - 2^\ell w_1$ satisfies the target modular equations (40) and (41), thereby completing the proof. □

It should be pointed out that these modular equations cannot be further relaxed. If we weaken the Schur product condition $w \cdot (x_i * x_j) = 0 \pmod{2^\ell}$ to modulo $2^{\ell-1}$, the expansion of f_1 in (43) gains an additional residual term:

$$2^\ell \sum_{1 \le i < j \le k} a(i) a(j) \frac{w \cdot (x_i * x_j)}{2^{\ell-1}} \pmod 2.$$

The cross-terms $a(i)a(j)$ make this expression non-linear over C. Consequently, it cannot be expressed as a dot product with a fixed vector, which prevents the linear refinement of w. While this residual could be interpreted as a bi-linear map over $C \times C$, physically implementing such a map requires physical two-qubit CZ gates, destroying fault tolerance.

Furthermore, weakening the constraint $w \cdot x_i = 0 \pmod{2^{\ell+1}}$ to modulo-$2^{\ell-1}$ produces the residual term:

$$2^{\ell-1} \sum_{i=1}^{k} a(i) \frac{w \cdot x_i - A(i)}{2^{\ell-1}} \pmod 4.$$

Because arithmetic modulo 4 is not linear over C, we once again forfeit the linearity required to refine w.

APPENDIX C. PROOF OF LEMMA 3

Proof. We first verify sufficiency. For any non-negative integer p and integer vector w, the operator U defined in (5) is precisely the transversal Z-rotation

$$U = \bigotimes_{i=1}^{n} R_Z\left(\frac{\pi w(i)}{2^p}\right).$$

Conversely, consider an arbitrary dyadic transversal Z-rotation as described in Remark 2:

$$U = \bigotimes_{i=1}^{n} R_Z\left(\frac{\pi q_i}{2^{p_i}}\right).$$

Let $p_{\max} = \max\{p_i : i \in [n]\}$, and define the integer vector $w = (w(1), \ldots, w(n)) \in \mathbb{Z}^n$ by setting $w(i) = 2^{p_{\max}-p_i} q_i$. By construction, at least one component of w is odd—specifically, any w_i for which $p_i = p_{\max}$. For any computational basis state $|x\rangle$ with $x \in \mathbb{F}_2^n$, the action of U evaluates to

$$\begin{aligned} U|x\rangle &= \left(\prod_{i=1}^{n} \exp\Big\{\iota \frac{\pi}{2^{p_i}} q_i x_i\Big\}\right)|x\rangle \\ &= \exp\Big\{\iota \frac{\pi}{2^{p_{\max}}} (w \cdot x)\Big\}|x\rangle. \end{aligned}$$

Consequently, U_{P} admits the diagonal representation

$$U = \operatorname{diag}\left(\exp\Big\{\iota \frac{\pi}{2^{p_{\max}}} (w \cdot x)\Big\} : x \in \mathbb{F}_2^n\right).$$

To establish uniqueness, suppose there exist two such parameterizations:

$$U = \operatorname{diag}\left(\exp\Big\{\iota \frac{\pi}{2^p} (w_1 \cdot x)\Big\} : x \in \mathbb{F}_2^n\right) = \operatorname{diag}\left(\exp\Big\{\iota \frac{\pi}{2^q} (w_2 \cdot x)\Big\} : x \in \mathbb{F}_2^n\right), \tag{44}$$

where $p, q \geq 0$ are integers, and $w_1, w_2 \in \mathbb{Z}^n$ each contain at least one odd component. Let i be an index such that $w_1(i)$ is odd. Evaluating both the representations at the standard basis vector $x = e_i$ yields

$$\exp\Big\{\iota \frac{\pi}{2^p} w_1(i)\Big\} = \exp\Big\{\iota \frac{\pi}{2^q} w_2(i)\Big\}.$$

Because $w_1(i)$ is odd, phase equality mandates that $q \geq p$. A symmetric argument applied to an odd component of w_2 implies that $p \geq q$, forcing $p = q$. Substituting $p = q$ in the diagonal equality (44) and evaluating at $x = e_i$ for every index $i \in [n]$ demonstrates that

$$w_1(i) = w_2(i) \pmod{2^{p+1}},$$

which completes the proof. □

APPENDIX D. PROOF OF PROPOSITION 6

Proof. By Eq. (10) in the proof of Theorem 5, the physical gate $U(p, w)$ realizes the logical gate U_{L} if and only if $p \geq \ell$ and for all $x \in C_2$ and $a \in \mathbb{F}_2^k$, the following condition holds:

$$w \cdot (x \oplus y_a) = 2^{p-\ell} \left(\sum_{1 \leq i_1 < \cdots < i_m \leq k} A(i_1, \ldots, i_m) a(i_1) \cdots a(i_m) \right) \pmod{2^{p+1}}. \tag{45}$$

We now show that condition (45) implies the stricter bound $p \geq \ell + m - 1$. We prove by contradiction, assuming $p \leq \ell + m - 2$. Because the addressability tensor A possesses at least one odd component, we may permute the indices such that $A(1, \ldots, m)$ is odd.

Let $\{y_1, \ldots, y_k\}$ denote the basis for the coset space C_1/C_2. For any non-empty subset $S \subseteq [m-1]$, setting $x = 0$ and $a = \bigoplus_{i \in S} e_i$ in condition (45), we obtain

$$w \cdot y_a = w \cdot \left(\bigoplus_{i \in S} y_i \right) = 0 \pmod{2^{p+1}}. \tag{46}$$

As a consequence of (46), for any non-empty subset $\{i_1, \ldots, i_r\} \subseteq [m-1]$, it follows that

$$(-2)^{r-1} \left(w \cdot (y_{i_1} * \cdots * y_{i_r}) \right) = 0 \pmod{2^{p+1}}. \tag{47}$$

Applying (47) to the evaluation at $a = \bigoplus_{i=1}^{m} e_i$, we deduce that

$$w \cdot y_a = (-2)^{m-1} \left(w \cdot (y_1 * \cdots * y_m) \right) \pmod{2^{p+1}}. \tag{48}$$

On the other hand, directly evaluating condition (45) at $a = \bigoplus_{i=1}^{m} e_i$ (with $x = 0$) yields

$$w \cdot y_a = 2^{p-\ell} A(1, \ldots, m) \pmod{2^{p+1}}. \tag{49}$$

Combining (48) and (49), we can express this modular congruence as an integer equation. For some $c \in \mathbb{Z}$, we have

$$\begin{aligned} & 2^{p-\ell} A(1, \ldots, m) = (-2)^{m-1} \left(w \cdot (y_1 * \cdots * y_m) \right) + 2^{p+1} c \\ \implies & A(1, \ldots, m) = (-1)^{m-1} 2^{m-p+\ell-1} \left(w \cdot (y_1 * \cdots * y_m) \right) + 2^{\ell+1} c. \end{aligned}$$

Because we assumed $p \leq m + \ell - 2$, the exponent $m - p + \ell - 1$ is strictly positive. Furthermore, since $\ell \geq 0$, the exponent $\ell + 1$ is also strictly positive. Thus, the right-hand side of the equation is even, implying that $A(1, \ldots, m)$ must be an even integer. This directly contradicts our initial premise that $A(1, \ldots, m)$ is odd. We therefore conclude that $p \geq \ell + m - 1$. □

APPENDIX E. PROOF OF LEMMA 9

Proof. We begin by proving necessity. For any $i \in [k]$, evaluating (15) at $a = e_i$ yields (16):

$$w \cdot x_i = A(i) \pmod{2^{\ell+1}}. \tag{50}$$

We prove (17) by induction on $j \in [\ell+1]$, where the base case $j = 1$ is already established by (50). For the induction step, assume the result holds for all $1 \le p \le j < \ell+1$. Then, for any ordered indices $1 \le i_1 < \cdots < i_p \le k$, if $p = 1$ we have (50) and if $p \ge 2$ we have

$$w \cdot \left(x_{i_1} * \cdots * x_{i_p}\right) = 0 \pmod{2^{\ell-p+2}}. \tag{51}$$

Now, for any ordered indices $1 \le i_1 < \cdots < i_{j+1} \le k$, evaluating (15) at $a = \bigoplus_{p=1}^{j+1} e_{i_p}$ yields

$$w \cdot \left(\bigoplus_{p=1}^{j+1} x_{i_p}\right) = \sum_{p=1}^{j+1} A(i_p) \pmod{2^{\ell+1}}. \tag{52}$$

By expanding the binary addition, the left-hand side of (52) can be rewritten as

$$w \cdot \left(\bigoplus_{p=1}^{j+1} x_{i_p}\right) = \sum_{p=1}^{j+1} (-2)^{p-1} \sum_{1 \le q_1 < \cdots < q_p \le j+1} w \cdot \left(x_{i_{q_1}} * \cdots * x_{i_{q_p}}\right) \pmod{2^{\ell+1}}.$$

Applying the induction hypothesis (50) and (51), we obtain

$$w \cdot \left(\bigoplus_{p=1}^{j+1} x_{i_p}\right) = \sum_{p=1}^{j+1} A(i_p) + (-2)^j w \cdot \left(x_{i_1} * \cdots * x_{i_{j+1}}\right) \pmod{2^{\ell+1}}. \tag{53}$$

Equating (52) and (53), we finally obtain

$$w \cdot \left(x_{i_1} * \cdots * x_{i_{j+1}}\right) = 0 \pmod{2^{\ell-j+1}},$$

completing the induction and establishing necessity. To prove sufficiency, we begin by expanding the binary addition on the left-hand side of (15) into an integer sum:

$$w \cdot \left(\bigoplus_{i=1}^{k} a(i)x_i\right) = \sum_{j=1}^{k} (-2)^{j-1} \left(\sum_{1 \le i_1 < \cdots < i_j \le k} \left(\prod_{p=1}^{j} a(i_p)\right) w \cdot (x_{i_1} * \cdots * x_{i_j})\right).$$

Because we evaluate this expression modulo $2^{\ell+1}$, any term in the outer summation with $j \ge \ell+2$ contains a coefficient of $(-2)^{j-1}$, which is a multiple of $2^{\ell+1}$ and therefore vanishes. Thus, the expansion strictly truncates to the first $\ell+1$ terms. Substituting (16) and (17) into these remaining terms causes all higher-order Schur products ($2 \le j \le \ell+1$) to vanish modulo $2^{\ell+1}$ as well, leaving only the linear terms:

$$\begin{aligned} w \cdot \left(\bigoplus_{i=1}^{k} a(i)x_i\right) &= \sum_{i=1}^{k} a(i)A(i) \pmod{2^{\ell+1}} \\ &= w_H(a * A) \pmod{2^{\ell+1}}. \end{aligned}$$

This completes the proof. □

APPENDIX F. PROOF OF LEMMA 7

Proof. The block length and logical dimension of the derived CSS code follow directly from the construction. To establish the minimum distance, we begin by lower-bounding the X-distance. For any $x \in C_1 \setminus C_2$, we can decompose x as $x = \widehat{x} + y$, where $\widehat{x} \in C_2$ and $y \in C_1/C_2$ represents a non-zero coset element. Partitioning these vectors according to their primary and appended coordinates, we write $\widehat{x} = (x', \widetilde{x})$ and $y = (y', \widetilde{y})$, where $x' \in C_2'$ and $y' \in C_1'/C_2'$. Because the appended submatrix $\begin{bmatrix} G_1^{(1)} & \cdots & G_1^{(m)} \end{bmatrix}$ merely extends the linearly independent rows of $G_{C_1'/C_2'}$, the non-zero condition on y enforces that $y' \neq 0$. Consequently, the weight of x is lower-bounded by the X-distance of the primary CSS code, yielding

$$d_X = d_{\min}(C_1 \setminus C_2) \geq d_{\min}(C_1' \setminus C_2') \geq d'.$$

Next, we turn to the Z-distance, d_Z. By the block-diagonal structure of G_{C_2}, the dual code $C_2^\perp$ decomposes into a direct sum of its constituent spaces:

$$C_2^\perp = (C_2')^\perp \oplus (C_2^{(1)})^\perp \oplus \cdots \oplus (C_2^{(m)})^\perp.$$

Consequently, we obtain the lower bound

$$d_Z = d_{\min}(C_2^\perp \setminus C_1^\perp) \geq d_{\min}(C_2^\perp) \geq \min\{d_{\min}((C_2')^\perp), d_{\min}((C_2^{(1)})^\perp), \ldots, d_{\min}((C_2^{(m)})^\perp)\}.$$

This completes the proof. □

REFERENCES

[1] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," *Phys. Rev. A*, vol. 52, pp. R2493(R)–R2496(R), Oct 1995.

[2] D. Gottesman, "Stabilizer codes and quantum error correction," 1997. arXiv:quant-ph/9705052.

[3] B. Eastin and E. Knill, "Restrictions on transversal encoded quantum gate sets," *Phys. Rev. Lett.*, vol. 102, p. 110502, Mar 2009.

[4] T. Tansuwannont, T. Chan, and R. Takagi, "Construction of the full logical Clifford group for high-rate quantum Reed-Muller codes using only transversal and fold-transversal gates," 2026. arXiv:2602.09788.

[5] N. P. Breuckmann and S. Burton, "Fold-Transversal Clifford Gates for Quantum Codes," *Quantum*, vol. 8, p. 1372, June 2024.

[6] T. Tansuwannont, Y. Takada, and K. Fujii, "Clifford gates with logical transversality for self-dual CSS codes," 2025. arXiv:2503.19790.

[7] S. Jain and V. Albert, "Transversal Clifford and T-gate codes of short length and high distance," *IEEE Journal on Selected Areas in Information Theory*, vol. 6, pp. 127–137, 2025.

[8] N. Sloane, F. MacWilliams, and J. Thompson, "Good self dual codes exist," *Discrete Mathematics*, vol. 3, pp. 153–162, 1972.

[9] C. Martinez-Perez and W. Willems, "Self-dual doubly even 2-quasi-cyclic transitive codes are asymptotically good," *IEEE Trans. Inf. Theor.*, vol. 53, p. 4302–4308, Nov. 2007.

[10] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. USA: Cambridge University Press, 10th ed., 2011.

[11] E. T. Campbell and C. Vuillot, "Roads towards fault-tolerant universal quantum computation," *Nature*, vol. 549, pp. 172–179, 2017.

[12] A. Krishna and J. Tillich, "Towards low overhead magic state distillation," *Phys. Rev. Lett.*, vol. 123, 2019.

[13] J. T. Anderson, G. Duclos-Cianci, and D. Poulin, "Fault-tolerant conversion between the steane and Reed-Muller quantum codes," *Phys. Rev. Lett.*, vol. 113, p. 080501, Aug 2014.

[14] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[15] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, (New York, NY, USA), p. 212–219, Association for Computing Machinery, 1996.

[16] M. A. Webster, A. O. Quintavalle, and S. D. Bartlett, "Transversal diagonal logical operators for stabiliser codes," *New Journal of Physics*, vol. 25, p. 103018, Oct 2023.

[17] J. T. Anderson and T. Jochym-O'Connor, "Classification of transversal gates in qubit stabilizer codes," *Quantum Info. Comput.*, vol. 16, p. 771–802, July 2016.

[18] J. Hu, Q. Liang, and R. Calderbank, "Designing the quantum channels induced by diagonal gates," *Quantum*, vol. 6, p. 802, 2022.

[19] J. Hu, Q. Liang, and R. Calderbank, "Climbing the diagonal Clifford hierarchy," *Academia Quantum*, vol. 2, 2025.

[20] J. Hu, Q. Liang, and R. Calderbank, "Divisible codes for quantum computation," 2022. arXiv:2204.13176.

[21] J. Haah, "Towers of generalized divisible quantum codes," *Phys. Rev. A*, vol. 97, Apr. 2018.

[22] H. Chen, C. Yu, Y. Chen, C. Xie, Z. Sun, and S. Zhu, "Optimal, best-known divisible binary codes and related quantum codes," *IEEE Transactions on Information Theory*, 2026.

[23] K. S. M. Reddy and N. Kashyap, "Asymptotically good CSS codes that realize the logical transversal Clifford group fault-tolerantly," 2026. arXiv:2601.08568.

[24] N. Rengaswamy, R. Calderbank, M. Newman, and H. Pfister, "On optimality of CSS codes for transversal T," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, pp. 499–514, 2020.

[25] N. Rengaswamy, R. Calderbank, M. Newman, and H. D. Pfister, "Classical coding problem from transversal T gates," in *2020 IEEE International Symposium on Information Theory (ISIT)*, p. 1891–1896, IEEE Press, 2020.

[26] S. Bravyi and J. Haah, "Magic-state distillation with low overhead," *Phys. Rev. A*, vol. 86, p. 052329, Nov 2012.

[27] E. Berardini, R. Dastbasteh, J. Martinez, S. Jain, and O. Larrarte, "Asymptotically good CSS-T codes and a new construction of triorthogonal codes," *IEEE Journal on Selected Areas in Information Theory*, vol. 6, pp. 189–198, 2025.

[28] E. Camps-Moreno, H. López, G. Matthews, D. Ruano, R. San-José, and I. Soprunov, "An algebraic characterization of binary CSS-T codes and cyclic CSS-T codes for quantum fault tolerance," *Quantum Information Processing*, vol. 23, p. 230, 2024.

[29] M. B. Hastings and J. Haah, "Distillation with sublogarithmic overhead," *Phys. Rev. Lett.*, vol. 120, p. 050504, Jan 2018.

[30] J. Bolkema, E. Andrade, T. Dexter, H. Eggers, V. L. Fisher, L. Szramowsky, and F. Manganiello, "CSS-T codes from Reed-Muller codes," *IEEE Journal on Selected Areas in Information Theory*, vol. 6, pp. 199–204, 2025.

[31] J. Haah and M. B. Hastings, "Codes and protocols for distilling T, controlled-S, and Toffoli gates," *Quantum*, vol. 2, p. 71, 2018.

[32] E. Camps-Moreno, H. H. López, G. L. Matthews, and E. McMillon, "Toward quantum CSS-T codes from sparse matrices," 2024. arXiv:2406.00425.

[33] E. Berardini, A. Caminata, and A. Ravagnani, "Structure of CSS and CSS-T quantum codes," *Des. Codes Cryptogr.*, vol. 92, May 2024.

[34] A. Baldelli, O. A. Mostad, H.-Y. Lin, E. Rosnes, and M. Battaglioni, "On constructing and decoding quantum triorthogonal codes," 2026. arXiv:2605.24519.

[35] J. J. Postema, F. Conca, and A. Ravagnani, "CSS-T codes over binary extension fields and their physical foundations," 2025. arXiv:2507.17611.

[36] Şeyma Bodur, F. Hernando, E. Martínez-Moro, and D. Ruano, "The schur product of evaluation codes and its application to CSS-T quantum codes and private information retrieval," 2026. arXiv:2505.10068.

[37] E. Camps-Moreno, H. H. López, G. L. Matthews, D. Ruano, R. San-José, and I. Soprunov, "Binary triorthogonal and CSS-T codes for quantum error correction," 2024.

[38] L. Golowich and V. Guruswami, "Asymptotically good quantum codes with transversal non-clifford gates," in *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, p. 707–717, Association for Computing Machinery, 2025.

[39] Q. T. Nguyen, "Good binary quantum codes with transversal CCZ gate," in *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, p. 697–706, Association for Computing Machinery, 2025.

[40] A. Wills, M. Hsieh, and H. Yamasaki, "Constant-overhead magic state distillation," *Nat. Phys.*, vol. 21, pp. 1842–1846, 2025.

[41] L. Golowich and T.-C. Lin, "Quantum LDPC codes with transversal non-clifford gates via products of algebraic codes," 2024. arXiv:2410.14662.

[42] S. J. S. Tan, Y. Hong, T.-C. Lin, M. J. Gullans, and M.-H. Hsieh, "Single-shot universality in quantum LDPC codes via code-switching," 2025. arXiv:2510.08552.

[43] T. Jochym-O'Connor and T. J. Yoder, "Four-dimensional toric code with non-clifford transversal gates," *Phys. Rev. Res.*, vol. 3, p. 013118, Feb 2021.

[44] O. Elishco and I. Tamo, "Quantum codes with transversal CCZ gates and sublinear Z-stabilizers," 2026. arXiv:2606.22472.

[45] R. Tiew and N. P. Breuckmann, "Copy-cup gates in tensor products of group algebra codes," 2026. arXiv:2602.23307.

[46] L. Golowich and V. Guruswami, "Near-asymptotically-good quantum codes with transversal CCZ gates and sublinear-weight parity-checks," 2025. arXiv:2510.06798.

[47] V. Menon, J. P. Bonilla Ataides, R. Mehta, A. Gu, D. B. Tan, and M. D. Lukin, "Magic tricycles: Efficient magic-state generation with finite block-length quantum LDPC codes," *Phys. Rev. X*, vol. 16, p. 021014, Apr 2026.

[48] A. Jacob, C. McLauchlan, and D. E. Browne, "Single-shot decoding and fault-tolerant gates with trivariate tricycle codes," 2026. arXiv:2508.08191.

[49] Y. Li, Z. Li, Z.-W. Liu, and Q. T. Nguyen, "Poincaré duality and multiplicative structures on quantum codes," 2025. arXiv:2512.21922.

[50] S. Bravyi and A. Cross, "Doubled color codes," 2015. arXiv:1509.03239.

[51] R. Dastbasteh, R. M. Otxoa, P. M. Crespo, and J. E. Martinez, "Quantum codes with arbitrary Z-rotation logical gates and applications to fault-tolerant code switching," 2026. arXiv:2608.11160.

[52] H. Bombín, "Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes," *New Journal of Physics*, vol. 17, p. 083002, aug 2015.

[53] A. Kubica and M. E. Beverland, "Universal transversal gates with color codes: A simplified approach," *Phys. Rev. A*, vol. 91, p. 032330, Mar 2015.

[54] B. J. Brown, "Color code with a logical control-S gate using transversal T rotations," *Phys. Rev. Lett.*, vol. 135, p. 070602, Aug 2025.

[55] A. Paetznick and B. W. Reichardt, "Universal fault-tolerant quantum computation with only transversal gates and error correction," *Phys. Rev. Lett.*, vol. 111, p. 090505, Aug 2013.

[56] A. Kubica, B. Yoshida, and F. Pastawski, "Unfolding the color code," *New Journal of Physics*, vol. 17, p. 083026, aug 2015.

[57] M. Vasmer and D. E. Browne, "Three-dimensional surface codes: Transversal gates and fault-tolerant architectures," *Phys. Rev. A*, vol. 100, p. 012312, Jul 2019.

[58] G. Zhu, "A topological theory for qLDPC: non-clifford gates and magic state fountain on homological product codes with constant rate and beyond the $N^{1/3}$ distance barrier," 2025. arXiv:2501.19375.

[59] G. Zhu, S. Sikander, E. Portnoy, A. W. Cross, and B. J. Brown, "Non-clifford and parallelizable fault-tolerant logical gates on constant and almost-constant rate homological quantum low-density parity-check codes via higher symmetries," *PRX Quantum*, vol. 6, p. 040361, Dec 2025.

[60] E. Camps-Moreno, H. H. López, G. L. Matthews, N. Rengaswamy, and R. San-José, "Transversal gates for quantum CSS codes," 2026. arXiv:2601.21514.

[61] J. Guyot and S. Jaques, "On the Addressability Problem on CSS Codes," *Quantum*, vol. 10, p. 2145, June 2026.

[62] A. Patra and A. Barg, "Targeted Clifford logical gates for hypergraph product codes," *Quantum*, vol. 9, p. 1842, Aug. 2025.

[63] A. O. Quintavalle, P. Webster, and M. Vasmer, "Partitioning qubits in hypergraph product codes to implement logical gates," *Quantum*, vol. 7, p. 1153, Oct. 2023.

[64] Z. He, V. Vaikuntanathan, A. Wills, and R. Y. Zhang, "Quantum codes with addressable and transversal non-clifford gates," 2025. arXiv:2502.01864.

[65] Z. He, V. Vaikuntanathan, A. Wills, and R. Y. Zhang, "Asymptotically good quantum codes with addressable and transversal non-clifford gates," 2025. arXiv:2507.05392.

[66] V. Guémard, "Good quantum codes with addressable and parallelizable transversal non-clifford gates," 2025. arXiv:2510.19809.

[67] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd ed., 2009.

[68] A. M. Steane, "Simple quantum error-correcting codes," *Phys. Rev. A*, vol. 54, pp. 4741–4751, Dec 1996.

[69] A. Ashikhmin, S. Litsyn, and M. A. Tsfasman, "Asymptotically good quantum codes," *Phys. Rev. A*, vol. 63, p. 032311, Feb 2001.

[70] J. Ax, "Zeroes of polynomials over finite fields.," *American Journal of Mathematics*, vol. 86, p. 255, 1964.

[71] K. Betsumiya and A. Munemasa, "On triply even binary codes," *Journal of the London Mathematical Society*, vol. 86, 2012.