

Co-Designing Quantum Codes with Transversal Diagonal Gates via Multi-Agent Systems

Xi He¹, Sirui Lu², Bei Zeng^{1*}

1. Department of Physics, The University of Texas at Dallas, Richardson, Texas 75080, USA
2. Max-Planck-Institut für Quantenoptik, Garching bei München - 85748, Germany

October 24, 2025

Abstract

We present a multi-agent, human-in-the-loop workflow that co-designs quantum codes with prescribed transversal diagonal gates. It builds on the Subset-Sum Linear Programming (SSLP) framework (arXiv:2504.20847), which partitions basis strings by modular residues and enforces Z-marginal Knill-Laflamme (KL) equalities via small LPs. The workflow is powered by GPT-5 and implemented within TeXRA (<https://texra.ai>)—a multi-agent research assistant platform that supports an iterative tool-use loop agent and a derivation-then-edit workflow reasoning agent. We work in a L^AT_EX–Python environment where agents reason, edit documents, execute code, and synchronize their work to Git/Overleaf. Within this workspace, three roles collaborate: a Synthesis Agent formulates the problem; a Search Agent sweeps/screens candidates and exactifies numerics into rationals; and an Audit Agent independently checks all KL equalities and the induced logical action. As a first step we focus on distance $d = 2$ with nondegenerate residues. For code dimension $K \in \{2, 3, 4\}$ and $n \leq 6$ qubits, systematic sweeps yield certificate-backed tables cataloging attainable cyclic logical groups—all realized by new codes—e.g., for $K = 3$ we obtain order 16 at $n = 6$. From verified instances, Synthesis Agent abstracts recurring structures into closed-form families and proves they satisfy the KL equalities for all parameters. It further demonstrates that SSLP accommodates residue degeneracy by exhibiting a new $((6, 4, 2))$ code implementing the transversal controlled-phase $\text{diag}(1, 1, 1, i)$. Overall, the workflow recasts diagonal-transversal feasibility as an analytical pipeline executed at scale, combining systematic enumeration with exact analytical reconstruction. It yields reproducible code constructions, supports targeted extensions to larger K and higher distances, and leads toward data-driven classification.

1 Introduction

AI systems are increasingly taking on novel research tasks [1, 2, 3, 4, 5], yet face challenges beyond current capabilities. Success requires identifying problems with the right structure: those requiring large-scale exploration of combinatorial search spaces where solutions are hard to generate but easy to verify. In mathematics and physics, many seminal solutions to difficult problems were found by constructing candidate solutions by hand, guided by intuition, and verifying them against constraints, often before a general theory was developed. While human intuition excels at identifying promising directions, enumeration at scale and the extraction of analytical patterns from large collections of examples remain bottlenecks. AI systems have long excelled at large-scale search—AlphaGo exploring game trees [6, 7], AlphaFold predicting structures [8], and automated theorem proving [9, 10]. Large language models [11] now add complementary capabilities: tool-use [12, 13, 14] (generating and executing domain code [15, 16]) and reasoning (deriving analytical theories from examples [17, 18]), though current systems still require substantial human supervision.

We implement a multi-agent system [19, 20, 21] and apply it to discover novel quantum error-correcting codes. The system is built around TeXRA [22], an agentic AI research assistant platform with GPT-5 [23] that enables file creation, code execution, and natural-language interaction within a working directory containing LaTeX formulations, Python code, and results. The working directory can be connected as a git repository to Overleaf for sharing among collaborators. The platform supports two distinct operational modes: a *tool-use loop* [24] where agents iteratively call functions—writing code, executing

*bei.zeng@utdallas.edu

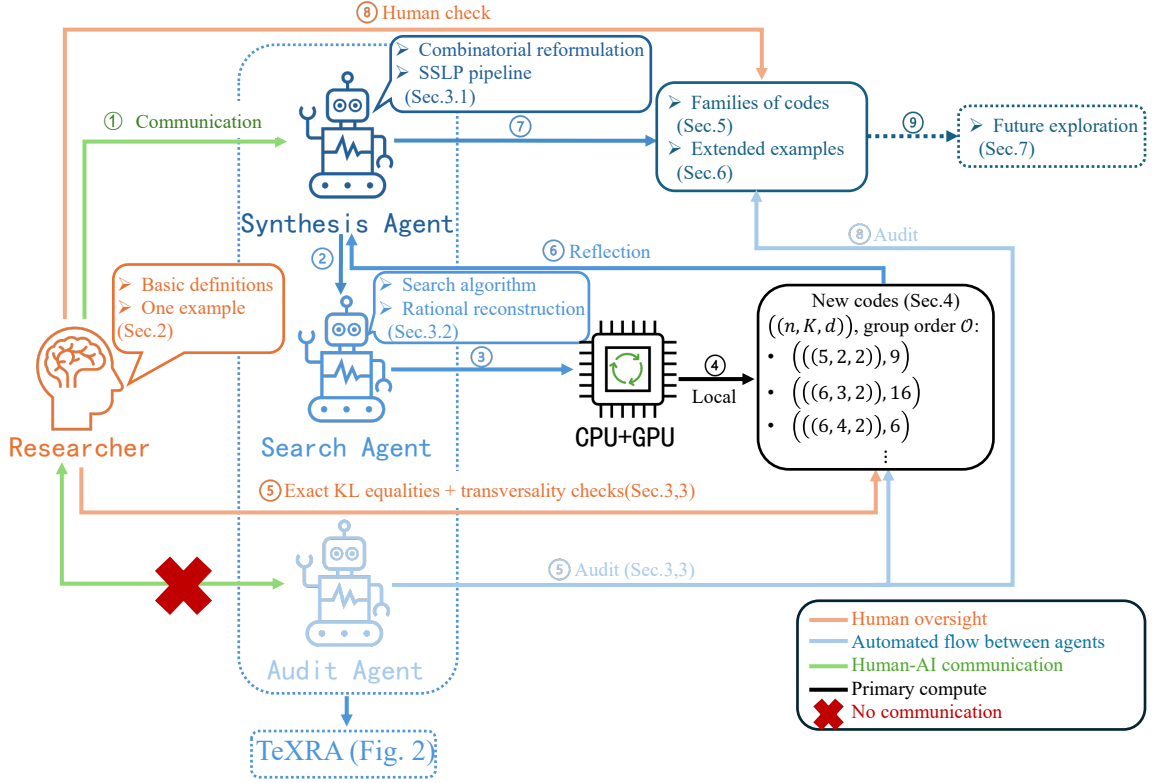


Fig. 1: Schematic diagram of the human-AI co-design workflow for quantum code discovery. The researcher seeds the system with basic definitions and a worked example (Sec. 2), reviews outputs, and provides checkpoints (orange). The Synthesis Agent deduces a combinatorial reformulation of the problem and proposes parameter templates (Sec. 3.1). Guided by these proposals, the Search Agent generates and executes search algorithms on a local workstation to enumerate candidate codes (black; Sec. 3.2). The resulting candidates are passed to the Audit Agent, which, along with the researcher, independently verifies that they satisfy the exact KL and transversal conditions (Sec. 3.3). Verified results are fed back to the Synthesis Agent to abstract general code families and construct extended examples (Secs. 4, 5). Legend: Blue arrows denote automated agent-to-agent flow; green arrows indicate human-AI communication; orange arrows mark human oversight; the red cross, **X**, highlights the intentional “no-communication” barrier that preserves the independence of the audit process.

searches, processing data—until reaching a stopping criterion; and a *derivation-then-edit workflow* where agents first expand mathematical derivations in an internal scratchpad, then produce edited LaTeX files with new content or critical annotations for human review. Three specialized agents operate under human orchestration in this shared environment (Fig. 1), combining tool-use and reasoning to address a feasibility-and-construction problem in quantum error correction: determining which transversal gate groups can arise for quantum codes with specified physical and logical parameters—a setting where large-scale search and analytical reasoning are both essential, and the results are straightforward to verify.

Quantum error correction protects information by encoding it into a K -dimensional subspace of a larger n -qubit system, enabling the detection or correction of errors [25, 26, 27, 28]. The central challenge of implementing logical operations without propagating errors is addressed by transversal gates—acting independently on each physical qubit—but no-go theorems forbid transversal gate sets for universal computation [29, 30] and impose stringent group-theoretic constraints [31, 32]. Beyond stabilizer codes [26, 33], nonadditive [34] constructions expand the design space for transversality; for example, codeword-stabilized (CWS) codes [35, 36, 37, 38, 39], permutation-invariant (PI) codes [40, 41, 42, 43], and beyond [44]. Small stabilizer subsystem codes have recently been systematically enumerated [45]. For nonadditive codes, transversal structure is strikingly rich: classical results already highlight nonadditive phenomena [46], and notable recent constructions include permutation-invariant codes realizing the binary icosahedral group $2I$ [47] and codes with higher-order diagonal phases [48]. Meanwhile, the Subset-Sum Linear Programming (SSLP) framework [49] reframes the *diagonal-transversal* question as congruence structure plus linear Z -marginals, enabling the discovery of many new codes with transversal properties,

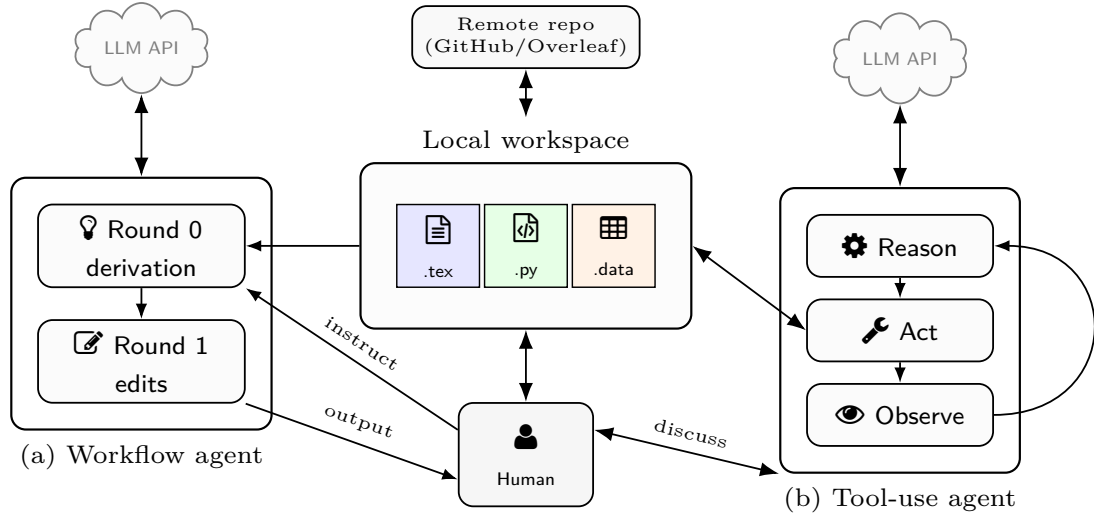


Fig. 2: Two types agent supported by TeXRA operating on a shared workspace (L^AT_EX files, Python scripts, data) synchronized to a remote repository (GitHub/Overleaf). Both communicate with GPT-5 via API. (a) *Derivation-then-edit workflow*: Round 0 derives, Round 1 extends reasoning (test-time scaling [17]) and generates L^AT_EX edits; human instructs and reviews output. (b) *Tool-use loop*: Iterative ReAct cycle (reason → act → observe) where act means writing/executing code; human discusses throughout. Bidirectional arrows show interactive communication.

revealing a far richer landscape of nonadditive codes than previously recognized, and underscoring a deeper connection between quantum error correction and the algebraic constraints on transversal gate groups. For quantum codes with parameters $((n, K, d))$ — n physical qubits, code dimension K , distance d —the key question is then: which transversal groups can arise and with what constructions?

We address this question for diagonal transversal gates—yielding abelian (typically cyclic) logical groups—using the recently proposed SSLP framework [49]. SSLP partitions computational basis strings into discrete groups using modular arithmetic: each logical basis state is assigned to one group, so a transversal diagonal gate—applying phases independently to each qubit—induces a predictable logical phase determined by the group label. Error-correction constraints (the Knill-Laflamme (KL) conditions [28] ensuring errors are detectable) then decompose into (i) structural requirements on group separation preventing bit-flip errors from creating unwanted interference between logical states, and (ii) constraints on how quantum amplitudes distribute within groups to ensure single-qubit measurement statistics match across logical states—the latter being linear in the squared amplitudes (probabilities). This structure—discrete choices (which groups?) plus linear feasibility (which probability distributions?)—makes systematic search tractable while still requiring both combinatorial enumeration and analytical pattern recognition.

The workflow proceeds as follows. The researcher provides initial definitions and a worked example in LaTeX, establishing notation and error-correction requirements (Sec. 2). The Synthesis Agent reads these definitions and proposes combinatorial formulations: which modular partitions satisfy the structural separation requirements, and which parameter families merit exploration. The Search Agent translates these proposals into executable Python code, performs large-scale enumeration on local computing resources, screens candidates using fast feasibility checks, solves linear constraint systems for $K \in \{2, 3, 4\}$ on $n \leq 6$ qubits, and converts numerical solutions to exact analytical form. Results pass through a no-communication barrier to the Audit Agent, which independently verifies every instance via dual numerical-analytical checks using separate implementations to confirm that KL conditions hold exactly and that transversal gates induce the claimed logical phases. Verified codes feed back to the Synthesis Agent, which identifies recurring structural patterns across successful instances, proposes closed-form infinite families, and derives proofs that these families satisfy error-correction constraints for all parameter values. Throughout this cycle, humans shape search strategies, validate agent-proposed generalizations, and perform targeted checks; agents execute the discovery and formal verification.

We focus initially on distance- $d = 2$ codes with nondegenerate residues (each logical state occupies a distinct modular class), where SSLP’s structure enables complete systematic exploration. For code dimensions $K \in \{2, 3, 4\}$ on up to $n = 6$ physical qubits, the Search Agent discovers new codes realizing cyclic gate orders ranging from 2 to 18—for instance, $K = 2$ codes achieving order 9 on $n = 5$ qubits and

order 18 on $n = 6$ -with explicit constructions specifying exact parameters, probability amplitudes, and phases (complete tables in Sec. 4). The Synthesis Agent extracts analytical understanding: identifying recurring structural patterns across verified instances, it proposes closed-form infinite families and derives proofs that these families satisfy error-correction constraints for all parameter values (Sec. 5). Beyond the baseline setting, we demonstrate the framework handles richer constraints: relaxing the nondegenerate-residue assumption, the agents construct a $((6, 4, 2))$ code realizing a controlled-phase gate $\text{diag}(1, 1, 1, i)$ where three logical states share a residue class (Sec. 6). This illustrates how the multi-agent approach scales from tractable regimes enabling classification to more complex settings where agents complement human insight. The workflow demonstrates that combining systematic search with analytical reasoning can address difficult combinatorial problems in mathematical physics, with large-scale enumeration revealing structure inaccessible to manual analysis alone.

The remainder of this paper is organized as follows. Section 2 presents the SSLP framework with a worked example. Section 3 details the three-agent architecture (Fig. 1), screening algorithms, rational reconstruction, and verification protocols. Section 4 reports discovered code tables. Section 5 presents agent-derived analytical families with proofs. Section 6 demonstrates extensions beyond nondegenerate residues with a $((6, 4, 2))$ controlled-phase example. Section 7 discusses contributions and our reflections on the co-design process.

2 Problem Co-Formulation: Agent Workspace, SSLP Framework, and the Seeding Example

2.1 Workspace Setup and Agent Workflows

Before detailing the mathematical framework, we describe the computational infrastructure that enables our human-AI collaboration. The workspace is built around TeXRA [22], a Visual Studio Code extension for academic research that integrates large language models into the local development environment. VS Code [50] is a widely-used source-code editor that provides file management, version control integration (git), and an extensible plugin architecture. TeXRA extends it with agentic capabilities for scientific writing and computation. The system operates locally: user interactions and file contents are sent as API requests to model providers (we used OpenAI’s GPT-5 API [23]), and the returned responses are parsed and displayed to the researcher or processed into structured outputs. We organized the workspace as a directory containing LaTeX source files, Python scripts, data files, and chat logs, which we connected as a git repository to Overleaf to allow collaborators to review agent-generated content and track changes in real time.

The platform supports two distinct operational modes (see Fig. 2 for an illustration). The *tool-use loop* implements the ReAct paradigm [24], alternating between reasoning and action in an interactive, open-ended process. Through a chat interface, the agent maintains a conversation history and can call functions [12], such as creating or editing files, executing Python scripts, reading outputs, or querying file contents. Each interaction cycle consists of (1) the agent reasoning about the next step, (2) selecting and invoking a tool with appropriate arguments, (3) the system executing the action and returning the results, and (4) the agent incorporating the results into its reasoning for the next step. This loop continues until the task is complete or the agent requests human guidance. We used this mode for exploratory tasks, such as proposing alternative problem formulations and iteratively refining search strategies.

The *derivation-then-edit workflow* is a fixed two-round process that combines chain-of-thought reasoning [51] with structured output generation and self-reflection [52]. We select input files and context materials through the interface. In the first round, the agent expands its intermediate reasoning steps in an internal scratchpad-deriving formulas, analyzing examples, proposing generalizations-without producing final output. In the second round, the agent generates new LaTeX content or critical annotations [53] based on its reasoning, wrapping the outputs in XML tags for structured parsing. The system uses regular expressions to extract these tagged segments and constructs a modified LaTeX document. We visualize the changes using `latexdiff` [54], which produces a markup showing additions and deletions, and review this diff in VS Code’s compare mode to accept or reject modifications. Modern LLMs support context windows exceeding 200K tokens [23, 13]-sufficient for multiple research papers-making them effective at synthesizing and reformulating existing mathematical frameworks.

Given these capabilities, we structure a multi-agent architecture [19, 20, 21] where specialized agents handle distinct subtasks under human orchestration. This separation provides *specialization* (each agent focuses on one function with appropriate prompting), *verification independence* (auditing agents do not

see intermediate artifacts from generation agents), and natural *checkpoints for human oversight* (we review proposals before committing to large computations and validate outputs to catch errors before they propagate). The specific agent roles and their implementation are detailed in Sec. 3.

2.2 Preliminaries

We now present the mathematical foundation that establishes our notation and the error-correction framework. These preliminaries were provided to the agents as seed input, partially synthesized from Ref. [49] using the derivation-then-edit workflow described above.

Notation. We work on an n -qubit system with computational basis $\{|x\rangle : x \in \{0,1\}^n\}$, where $x = (x_1, \dots, x_n)$ and $|x\rangle \equiv |x_1\rangle \otimes \dots \otimes |x_n\rangle$. The Hamming weight is $\text{wt}(x) = \sum_i x_i$, and the Hamming distance is $d_H(x, y) = \text{wt}(x \oplus y)$. The single-qubit Pauli operators are

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

where Z_i acts on a basis state as $Z_i |x\rangle = (-1)^{x_i} |x\rangle$. We denote code parameters by $((n, K, d))$ [25], for n physical qubits, a code dimension of K , and a distance of d . We use the standard notation $[n] := \{1, \dots, n\}$ for index sets.

A quantum code with distance d encodes information into a K -dimensional subspace spanned by orthonormal logical states $\{|j_L\rangle\}_{j=0}^{K-1}$. The KL conditions [28] ensure error detectability: for a set $\mathcal{E}$ of detectable errors, $\langle j_L | E^\dagger E' | j'_L \rangle = 0$ when $j \neq j'$, and $\langle j_L | E^\dagger E' | j_L \rangle = \lambda_{E, E'}$ is independent of j .

For distance $d = 2$, we take the set of detectable errors to be $E_D = \{X_i, Y_i, Z_i : i \in [n]\}$ (all single-qubit Paulis). The KL conditions [28] simplify to

$$\begin{aligned} \langle j_L | P_i | j'_L \rangle &= 0 \text{ for } j \neq j', \\ \langle j_L | P_i | j_L \rangle &= \lambda_{P_i} \text{ for all } j \in \{0, \dots, K-1\}, \end{aligned} \quad (1)$$

for all $P_i \in \{X_i, Y_i, Z_i\}$, where the scalars λ_{P_i} are independent of the logical state j . Writing a logical state as $|j_L\rangle = \sum_x a_{j,x} |x\rangle$ with probabilities $p_{j,x} = |a_{j,x}|^2$, the diagonal constraints for Z_i become

$$\sum_x (1 - 2x_i) p_{j,x} = t_i \text{ for all } i \in [n], j = 0, \dots, K-1, \quad (2)$$

for some site-wise constants $t_i \in \mathbb{R}$. The constraints for X_i and Y_i involve cross-terms of the form $a_{j,x}^* a_{j', x \oplus e_i}$ between Hamming-1 neighbors, which couple the amplitudes nonlinearly.

Subset-sum classes and modular inner product. Fix a modulus $m \in \mathbb{Z}_{>0}$ and a weight vector $\mathbf{w} = (w_1, \dots, w_n) \in (\mathbb{Z}_m)^n$. We define the modular inner product as

$$\langle \mathbf{w}, x \rangle \equiv \sum_{i=1}^n w_i x_i \pmod{m},$$

and the corresponding residue classes as

$$C_{S_j}(\mathbf{w}) := \{x \in \{0,1\}^n : \langle \mathbf{w}, x \rangle \equiv S_j \pmod{m}\}, \text{ for } j = 0, \dots, K-1. \quad (3)$$

Throughout this paper, we require each logical basis state to be supported on a single such class: $\text{supp}(|j_L\rangle) \subseteq C_{S_j}(\mathbf{w})$ for a set of residues $\mathbf{S} = (S_0, \dots, S_{K-1}) \in (\mathbb{Z}_m)^K$. We take $S_0 = 0$ without loss of generality. Let $C := \bigcup_j C_{S_j}(\mathbf{w})$ denote the classical union support of the code, and let $d(C) := \min_{x \neq y \in C} d_H(x, y)$ be its minimum distance.

Transversal diagonals and logical order. We consider the phase gate

$$Z(\theta) = \text{diag}(1, e^{i\theta}),$$

and define the transversal operator

$$U(\mathbf{w}, m) := \bigotimes_{i=1}^n Z\left(\frac{2\pi w_i}{m}\right), \quad U(\mathbf{w}, m) |x\rangle = \omega_m^{\langle \mathbf{w}, x \rangle} |x\rangle, \text{ where } \omega_m = e^{2\pi i/m}. \quad (4)$$

If $\text{supp}(|j_L\rangle) \subseteq C_{S_j}(\mathbf{w})$, then $U(\mathbf{w}, m)$ acts diagonally on the logical basis:

$$U(\mathbf{w}, m) |j_L\rangle = \omega_m^{S_j} |j_L\rangle, \quad \bar{U} = \text{diag}(\omega_m^{S_0}, \dots, \omega_m^{S_{K-1}}), \quad (5)$$

The (projective) order of the induced cyclic logical group is

$$\mathcal{O} = \frac{m}{\gcd(m, S_1, \dots, S_{K-1})}. \quad (6)$$

Using the rotation gate $R_Z(\theta) = e^{-i\theta Z/2}$ instead only introduces a global phase, leaving relative logical phases invariant; however, the absolute order of the logical cyclic group can double [49].

SSLP [49] Given parameters $(n, K, m, \mathbf{w}, \mathbf{S})$, where $\mathbf{w} = (w_1, \dots, w_n) \in (\mathbb{Z}_m)^n$ and $\mathbf{S} = \{S_0, \dots, S_{K-1}\} \subset \mathbb{Z}_m$ (with $S_0 = 0$), the SSLP framework consists of three main steps:

Step 1: Determining Logical-State Support Subsets. Compatibility with the transversal gate $U(\mathbf{w}, m)$ forces each logical basis state $|j_L\rangle$ to be supported exclusively on a single residue class:

$$\text{supp}(|j_L\rangle) \subseteq C_{S_j}(\mathbf{w}) := \{x \in \{0, 1\}^n : \langle \mathbf{w}, x \rangle \equiv S_j \pmod{m}\}.$$

If the residues $\{S_j\}$ are distinct, the supports $C_{S_j}(\mathbf{w})$ are disjoint, which ensures orthogonality and dramatically simplifies the search.

Step 2: Linear-Programming Filter from Z-type KL Conditions. We introduce non-negative probabilities $p_{j,x} = |a_{j,x}|^2$ for each logical state j , defined on its support $C_{S_j}(\mathbf{w})$ and normalized such that $\sum_{x \in C_{S_j}} p_{j,x} = 1$. The single-site Z -marginal equalities require the existence of site parameters $t_i \in \mathbb{R}$ such that

$$\sum_{x \in C_{S_j}(\mathbf{w})} (1 - 2x_i) p_{j,x} = t_i, \quad \forall i = 1, \dots, n, \quad \forall j = 0, \dots, K-1,$$

along with the non-negativity constraints $p_{j,x} \geq 0$. This is a linear feasibility program; any parameter set $(\mathbf{w}, \mathbf{S})$ for which this program is infeasible can be discarded immediately.

Step 3: Solving for the remaining KL conditions. With supports fixed (Step 1) and Z -type marginals feasible (Step 2), we now solve for complex amplitudes $a_{j,x}$ so that the *full* KL equalities hold for errors beyond the Z -only case. This can be done by minimizing a differentiable KL loss on the Stiefel manifold as shown in [49]. Because X (and Y) errors map C_{S_j} into $C_{S_j \pm w_i}$, these terms couple distinct residue blocks, making the problem nonconvex and globally coupled. In practice, this stage is the computational bottleneck: solutions are typically numerical, hard to verify analytically, and difficult to scale to larger n .

2.3 A concrete example in detail

We illustrate the SSLP pipeline with a detailed $((5, 2, 2))$ example. The key simplification is to impose the classical union distance $d(C) = 2$ (with $C = C_{S_0}(\mathbf{w}) \cup C_{S_1}(\mathbf{w})$), which forbids any Hamming-1 neighbors anywhere in C . Consequently, a single-qubit flip sends each $|x\rangle \in C$ to $|x \oplus e_i\rangle \notin C$, so for all i and all logical labels j, k ,

$$\langle j_L | X_i | k_L \rangle = \langle j_L | Y_i | k_L \rangle = 0.$$

Thus *both* the diagonal and off-diagonal KL terms for X/Y vanish identically; the Stiefel-manifold stage (SSLP Step 3) is bypassed, and the search collapses to the linear Z -marginal equalities, after which amplitudes admit rational solutions—the agent subsequently “exactified” the numerics via rational reconstruction, yielding certificate-backed analytical codes at scale.

Parameters.

$$n = 5, \quad \mathbf{w} = (1, 1, 2, 2, 2) \text{ (sorted nondecreasingly)}, \quad m = 7, \quad K = 2, \quad (S_0, S_1) = (0, 4).$$

Let $C_S(\mathbf{w}) := \{x \in \{0, 1\}^5 : \langle \mathbf{w}, x \rangle \equiv S \pmod{7}\}$, where $\langle \mathbf{w}, x \rangle \equiv \sum_{i=1}^5 w_i x_i \pmod{7}$. We impose the design condition

$$d(C) = 2, \quad \text{where } C := C_{S_0}(\mathbf{w}) \cup C_{S_1}(\mathbf{w}).$$

So there are no Hamming-1 neighbors anywhere in C ; hence all X_i/Y_i off-diagonal KL constraints are automatically satisfied.

Step 1: Subset-sum supports. The two residue classes are:

$$\begin{aligned} C_{S_0}(\mathbf{w}) &= C_0 = \{00000, 01111, 10111\}, \\ C_{S_1}(\mathbf{w}) &= C_1 = \{00011, 00101, 00110, 11001, 11010, 11100\}. \end{aligned}$$

(These are the basis strings; amplitudes will be introduced below.)

Step 2: Enforcing classical distance. The classical union $C = C_0 \cup C_1$ satisfies $d(C) = 2$. By enforcing this condition, we rule out any Hamming-1 neighbors within the code's support. This means no single-qubit X or Y flip can map a basis string from one logical state's support to another, causing all single-qubit (X/Y) off-diagonal terms between distinct logical states to vanish.

Step 3: Logical states (parameterized) We define the logical states with amplitudes supported on the designated residue classes:

$$\begin{aligned} |0_L\rangle &= \sum_{x \in C_0} a_{0,x} |x\rangle = \alpha_0 |00000\rangle + \alpha_1 |01111\rangle + \alpha_2 |10111\rangle, \\ |1_L\rangle &= \sum_{y \in C_1} a_{1,y} |y\rangle = \beta_0 |00011\rangle + \beta_1 |00101\rangle + \beta_2 |00110\rangle + \beta_3 |11001\rangle + \beta_4 |11010\rangle + \beta_5 |11100\rangle, \end{aligned} \quad (7)$$

with probabilities $p_{0,x_j} := |\alpha_j|^2$ for $x_j \in C_0$ and $p_{1,y_k} := |\beta_k|^2$ for $y_k \in C_1$, subject to the normalizations

$$\sum_{j=0}^2 p_{0,x_j} = 1, \quad \sum_{k=0}^5 p_{1,y_k} = 1.$$

Step 4: Transversal diagonal action. Define the transversal gate

$$U(\mathbf{w}, 7) := \bigotimes_{i=1}^5 Z\left(\frac{2\pi w_i}{7}\right), \quad U(\mathbf{w}, 7) |x\rangle = \omega_7^{\langle \mathbf{w}, x \rangle} |x\rangle, \quad \text{where } \omega_7 = e^{2\pi i/7}.$$

Because $\text{supp}(|j_L\rangle) \subseteq C_{S_j}(\mathbf{w})$, this gate acts diagonally on the logical basis:

$$U(\mathbf{w}, 7) |0_L\rangle = \omega_7^0 |0_L\rangle = |0_L\rangle, \quad U(\mathbf{w}, 7) |1_L\rangle = \omega_7^4 |1_L\rangle.$$

The induced logical gate is $\bar{U} = \text{diag}(1, \omega_7^4)$, which has order $7/\gcd(7, 4) = 7$.

Step 5: KL constraints from Z_i (SSLP linear stage). Since the $d(C) = 2$ condition ensures the X_i/Y_i constraints are met, only the diagonal Z_i constraints remain. These require

$$\langle 0_L | Z_i | 0_L \rangle = \langle 1_L | Z_i | 1_L \rangle, \quad \text{for } i = 1, \dots, 5,$$

which translates to a set of linear equations on the probabilities:

$$\sum_{x \in C_0} (1 - 2x_i) p_{0,x} = \sum_{y \in C_1} (1 - 2y_i) p_{1,y}, \quad \text{for } i = 1, \dots, 5.$$

In terms of the variables p_{0,x_j} and p_{1,y_k} , this yields the system:

$$\begin{aligned} (i=1) \quad & p_{0,x_0} + p_{0,x_1} - p_{0,x_2} = p_{1,y_0} + p_{1,y_1} + p_{1,y_2} - p_{1,y_3} - p_{1,y_4} - p_{1,y_5}, \\ (i=2) \quad & p_{0,x_0} - p_{0,x_1} + p_{0,x_2} = p_{1,y_0} + p_{1,y_1} + p_{1,y_2} - p_{1,y_3} - p_{1,y_4} - p_{1,y_5}, \\ (i=3) \quad & p_{0,x_0} - p_{0,x_1} - p_{0,x_2} = p_{1,y_0} - p_{1,y_1} - p_{1,y_2} + p_{1,y_3} + p_{1,y_4} - p_{1,y_5}, \\ (i=4) \quad & p_{0,x_0} - p_{0,x_1} - p_{0,x_2} = -p_{1,y_0} + p_{1,y_1} - p_{1,y_2} + p_{1,y_3} - p_{1,y_4} + p_{1,y_5}, \\ (i=5) \quad & p_{0,x_0} - p_{0,x_1} - p_{0,x_2} = -p_{1,y_0} - p_{1,y_1} + p_{1,y_2} - p_{1,y_3} + p_{1,y_4} + p_{1,y_5}. \end{aligned}$$

Step 6: Solve the LP equalities (one convenient family). Solving this system along with the two normalization constraints gives a family of solutions:

$$\begin{aligned} p_{0,x_0} &= \frac{3}{7}, & p_{0,x_1} &= p_{0,x_2} = \frac{2}{7}, \\ p_{1,y_0} &= \frac{1}{7} + p_{1,y_5}, & p_{1,y_1} &= \frac{1}{7} + p_{1,y_4}, & \text{where } p_{1,y_4}, p_{1,y_5} &\geq 0 \text{ and } p_{1,y_4} + p_{1,y_5} \leq \frac{2}{7}. \\ p_{1,y_2} &= \frac{3}{7} - p_{1,y_4} - p_{1,y_5}, & p_{1,y_3} &= \frac{2}{7} - p_{1,y_4} - p_{1,y_5}, \end{aligned}$$

Step 7: A canonical point (set $p_{1,y_4} = p_{1,y_5} = 0$). A simple choice from this family gives the explicit logical states:

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{3}{7}} |00000\rangle + \sqrt{\frac{2}{7}} |01111\rangle + \sqrt{\frac{2}{7}} |10111\rangle, \\ |1_L\rangle &= \sqrt{\frac{1}{7}} |00011\rangle + \sqrt{\frac{1}{7}} |00101\rangle + \sqrt{\frac{3}{7}} |00110\rangle + \sqrt{\frac{2}{7}} |11001\rangle. \end{aligned}$$

Step 8: Site expectations (shared by both logical states). For this solution (and indeed for the entire family), the expectation values of the Z_i operators are:

$$\langle Z_i \rangle = \begin{cases} \frac{3}{7}, & i = 1, 2, \\ -\frac{1}{7}, & i = 3, 4, 5, \end{cases}$$

which is consistent with the KL equalities. This instance realizes a $((5, 2, 2))$ code that admits the transversal diagonal gate $\bar{U} = \text{diag}(1, \omega_7^4)$ and satisfies all distance-2 KL conditions. The basic definitions and this single worked example are all that is required to seed the multi-agent system.

3 Multi-agent orchestration

Based on the problem definition and worked example from Section 2, the multi-agent system formalizes the search strategy and executes the discovery pipeline. The Synthesis Agent first analyzes the problem structure to deduce a combinatorial reformulation and a scalable SSLP workflow (Sec. 3.1). The Search Agent then implements this workflow, generating code to perform large-scale enumeration and exact rational reconstruction of candidate codes (Sec. 3.2). Finally, all discovered candidates are rigorously and independently verified by the Audit Agent and the human researcher to ensure correctness (Sec. 3.3).

3.1 Synthesis agent

As the primary interface with the researcher, the Synthesis Agent analyzes the initial definitions and the provided example to propose the following search strategies.

3.1.1 Combinatorial reformulation

First, the Synthesis Agent uses its LLM backend to recast the distance-2 construction in purely combinatorial terms. The key insight is that subset-sum residue classes can be used to simultaneously (i) enforce separation against single-bit flips and (ii) reduce the remaining Z -marginal constraints to a finite convex feasibility problem.

Residue separation implies distance two (screen). A key insight formulated by the agent is to use residue separation to satisfy the KL conditions for X and Y errors combinatorially. A single-bit flip at site i maps a string x to $x \oplus e_i$, which shifts its residue by $\pm w_i$ modulo m :

$$\langle \mathbf{w}, x \oplus e_i \rangle \equiv \langle \mathbf{w}, x \rangle \pm w_i \pmod{m}.$$

Thus, to prevent all Hamming-1 adjacencies between the supports of different logical states, we can simply forbid residue differences that coincide with any $\pm w_i$. This leads to the lightweight screening condition:

$$S_j - S_k \not\equiv \pm w_i \pmod{m} \quad \text{for every } i \in [n] \text{ and } j \neq k. \quad (8)$$

When this condition holds, the classical union support C has a minimum distance of $d(C) \geq 2$. As a result, all single-qubit X/Y off-diagonal terms between distinct logical states vanish without any need for phase engineering.

(If some coordinate is pinned within a class, Eq. (8) is sufficient but not necessary; an explicit $d(C)$ check may be substituted.)

Z marginals as a finite convex intersection. Define the sign vector associated with a binary string x as $v(x) := ((-1)^{x_1}, \dots, (-1)^{x_n}) \in \{\pm 1\}^n$. For each residue class $C_{S_j}(\mathbf{w})$, let $V_j := \{v(x) : x \in C_{S_j}(\mathbf{w})\}$ be the set of corresponding sign vectors. The set of all possible single-site Z -expectation vectors realizable by a state supported on class j is precisely the convex hull of V_j , denoted $\text{conv}(V_j) \subset [-1, 1]^n$. The distance-2 KL equalities for the Z_i operators are satisfied if and only if there exists a common expectation vector $q = (t_1, \dots, t_n)$ such that $q \in \bigcap_{j=0}^{K-1} \text{conv}(V_j)$. This is equivalent to finding probabilities $\{p_{j,x}\}$ that satisfy:

$$\sum_{x \in C_{S_j}(\mathbf{w})} (1 - 2x_i) p_{j,x} = t_i, \quad \forall i \in [n], \forall j, \quad \text{with} \quad \sum_{x \in C_{S_j}} p_{j,x} = 1 \text{ and } p_{j,x} \geq 0. \quad (9)$$

This reduction to a convex feasibility problem is the key simplification. Without residue separation, the KL conditions for X/Y errors would couple the amplitudes $a_{j,x}$ and their phases quadratically across different residue classes, leading to large and ill-conditioned nonlinear systems. By enforcing $d(C) \geq 2$, we transform this intractable nonlinear problem into the tractable linear program of Eq. (9).

Small supports via Carathéodory’s theorem. By Carathéodory’s theorem [55] in $\mathbb{R}^n$, if the intersection $\bigcap_j \text{conv}(V_j)$ is non-empty, then each class j admits a solution supported on at most $n + 1$ basis states. This guarantees that we can find sparse solutions. A convenient integer formulation of the feasibility problem is to find integer vectors u_j and a common integer vector t such that:

$$A_j u_j = t, \quad \mathbf{1}^\top u_j = L \leq n + 1, \quad u_j \in \mathbb{Z}_{\geq 0}^{|C_{S_j}|} \quad (\forall j), \quad (10)$$

where A_j is a matrix whose columns are the sign vectors $v(x)$ for $x \in C_{S_j}(\mathbf{w})$. Normalizing by L gives the rational solution $q = t/L$.

Fast no-go certificates. To efficiently prune the vast parameter space, the agent also formulates a method for generating fast no-go certificates. If $\bigcap_j \text{conv}(V_j) = \emptyset$, the parameter set is infeasible. This is often witnessed by a linear separator: if there exist $\alpha \in \mathbb{Z}^n$ and $\beta \in \mathbb{R}$ such that

$$\max_{x \in C_{S_j}} \alpha \cdot v(x) < \beta < \min_{x \in C_{S_k}} \alpha \cdot v(x) \quad \text{for some } j \neq k,$$

then the convex sets are strictly separated, and Eq. (9) is impossible. Computing $\alpha \cdot v(x)$ is extremely fast, allowing us to use such separators to reject large portions of the $(m, \mathbf{w}, \mathbf{S})$ search space before attempting a full LP solve.

Two illustrative patterns. *Homogeneous weights.* If $\mathbf{w} = (t, \dots, t)$ with $\gcd(t, m) = 1$, then $C_{S_j}(\mathbf{w})$ coincides with Hamming-weight layers modulo m :

$$C_{S_j}(\mathbf{w}) = \{x : \text{wt}(x) \equiv S_j t^{-1} \pmod{m}\}.$$

For small n and multiple distinct layers, the screen Eq. (8) typically forces an extreme layer (e.g. $\text{wt} = 0$ or n), which pins V_j to $\{\pm(1, \dots, 1)\}$ and often prevents a common q with the same parity $L \leq n + 1$ across the remaining layers. *Mixed weights (affine slices).* For many $\mathbf{w}$, the congruence fixes $\sum_{i \in I} x_i$ modulo m , so V_j lies in an affine slice $\{\mu \in [-1, 1]^n : \sum_{i \in I} \mu_i = c'_j\}$ with different constants c'_j . If these slices share no rational point with denominator $\leq n + 1$, a linear functional separates them, yielding an immediate no-go.

3.1.2 SSLP Pipeline

The Synthesis Agent formalizes the SSLP framework into the following operational pipeline.

Inputs/outputs. *Inputs:* Code parameters $((n, K))$, modulus m , weight vector $\mathbf{w} \in (\mathbb{Z}_m)^n$, and residues $\mathbf{S} \in (\mathbb{Z}_m)^K$. *Outputs:* (i) Residue-class supports $C_{S_j}(\mathbf{w})$ that pass the screening; (ii) rational probabilities $p_{j,x}$ that solve the LP; and (iii) the induced logical diagonal gate $\bar{U}$ and its order $\mathcal{O}$.

Step 1 - Build subset-sum supports. Compute the residue classes

$$C_{S_j}(\mathbf{w}) := \{x \in \{0, 1\}^n : \langle \mathbf{w}, x \rangle \equiv S_j \pmod{m}\}, \quad j = 0, \dots, K - 1,$$

and require that $C_{S_j}(\mathbf{w})$ is non-empty for all j . Let $C := \bigcup_j C_{S_j}(\mathbf{w})$.

Step 2 - Separation screen (eliminate X/Y off-diagonals). To guarantee that $\langle j_L | X_i | j'_L \rangle = \langle j_L | Y_i | j'_L \rangle = 0$ for all i and $j \neq j'$, we enforce one of the following conditions at construction time:

(2a) *Distance check:* Ensure the classical minimum distance of the union support satisfies

$$d(C) := \min\{d_H(x, y) : x \neq y, x, y \in C\} \geq 2.$$

This forbids any Hamming-1 neighbors across residue classes, so a single bit flip cannot map a basis string from the support of one logical state to that of another.

(2b) *Modular shift screen (fast sufficient rule):* For every site i and every pair of distinct logical states j, k , enforce

$$S_j - S_k \not\equiv \pm w_i \pmod{m}. \quad (11)$$

This is a lightweight combinatorial filter that is sufficient to guarantee $d(C) \geq 2$. We use it to prune the search space in large sweeps.

Under either condition, the KL constraints for X_i and Y_i are automatically satisfied.

Step 3 - Linear program for Z -marginals. Introduce non-negative variables $p_{j,x}$ for $x \in C_{S_j}(\mathbf{w})$, subject to the normalizations $\sum_{x \in C_{S_j}} p_{j,x} = 1$. Enforce the single-site Z_i equalities across all logical states by requiring the existence of site parameters $t_i \in \mathbb{R}$:

$$\sum_{x \in C_{S_j}(\mathbf{w})} (1 - 2x_i) p_{j,x} = t_i, \quad \forall i \in [n], \forall j \in \{0, \dots, K-1\}. \quad (12)$$

Solve this linear feasibility program to find a canonical solution.

Step 4 - Read out the induced transversal action. With $\text{supp}(|j_L\rangle) \subseteq C_{S_j}(\mathbf{w})$, the transversal diagonal gate

$$U(\mathbf{w}, m) = \bigotimes_{i=1}^n Z\left(\frac{2\pi w_i}{m}\right), \quad U(\mathbf{w}, m) |x\rangle = \omega_m^{\langle \mathbf{w}, x \rangle} |x\rangle,$$

acts on the code space as

$$U(\mathbf{w}, m) |j_L\rangle = \omega_m^{S_j} |j_L\rangle, \quad \overline{U} = \text{diag}(\omega_m^{S_0}, \dots, \omega_m^{S_{K-1}}),$$

with a logical cyclic order of

$$\mathcal{O} = \frac{m}{\gcd(m, S_1, \dots, S_{K-1})} \quad (\text{taking } S_0 = 0 \text{ wlog}).$$

Step 5 - Audit. Independently verify all KL equalities using exact arithmetic on the constructed states. This includes confirming that (i) the X/Y off-diagonals vanish (guaranteed by Step 2), (ii) the Z conditions are met (guaranteed by Step 3), and (iii) each logical state is properly normalized.

This distance-2 SSLP pipeline effectively separates the combinatorial aspects of the problem from the convex ones. Residue-class separation handles all X/Y off-diagonals at the level of supports, while the Z -marginal constraints reduce to a compact LP that yields exact rational solutions. The outcome is a fully auditable construction of a quantum code, along with its induced diagonal transversal gate and logical order $\mathcal{O}$.

3.2 Search agent

Guided by the combinatorial reformulation and the SSLP pipeline from the Synthesis Agent, the Search Agent generates code to enumerate and filter parameter sets $(m, \mathbf{w}, \mathbf{S})$ at scale and to solve the corresponding Z -only linear programs. The search process is implemented to run efficiently on a local computing workstation.

3.2.1 Search (enumeration & feasibility)

Search space and guards. To manage the combinatorial search space and avoid redundant enumeration, we enforce canonical representatives. The agent enumerates weight vectors $\mathbf{w}$ in nondecreasing order ($1 \leq w_1 \leq \dots \leq w_n \leq m-1$) and residue tuples $\mathbf{S}$ with $S_0 = 0$ and $1 \leq S_1 < \dots < S_{K-1}$. This canonicalization, combined with tracking equivalence classes under qubit permutations, ensures that each discovered code is genuinely distinct. Two lightweight guards are applied before any heavy computation: (i) an optional coprime filter on the residues S_j ; and (ii) the residue-shift screen

$$S_j \not\equiv S_k \pm w_i \pmod{m} \quad \text{for all } i \in [n], j \neq k, \quad (13)$$

which forbids all Hamming-1 adjacencies between the supports of different logical states.

Supports and union distance. For each parameter set $(\mathbf{w}, m, \mathbf{S})$ that passes the initial guards, we compute the residue classes $C_{S_j}(\mathbf{w})$ by evaluating $\langle \mathbf{w}, x \rangle \pmod{m}$ for all $x \in \{0, 1\}^n$. We reject any set for which a class is empty. We then enforce a distance of two on the union support:

$$C := \bigcup_{j=0}^{K-1} C_{S_j}(\mathbf{w}), \quad d(C) = \min_{\substack{x \neq y \\ x, y \in C}} d_H(x, y) = 2. \quad (14)$$

While Eq. (14) is the definitive check, the modular shift screen of Eq. (13) provides a faster sufficient condition that we prioritize in large-scale sweeps. Empirically, these screens remove the vast majority of candidates before any LP solve is attempted.

Z-only feasibility (LP). Let $p_{j,x}$ be non-negative, block-normalized probabilities on the support $C_{S_j}(\mathbf{w})$, such that $\langle j_L | Z_i | j_L \rangle = \sum_{x \in C_{S_j}(\mathbf{w})} p_{j,x} (1 - 2x_i)$. The KL equalities for all Z_i are satisfied if there exists a set of probabilities $\{p_{j,x}\}$ satisfying:

$$\begin{aligned} \sum_{x \in C_{S_0}(\mathbf{w})} p_{0,x} (1 - 2x_i) &= \sum_{y \in C_{S_j}(\mathbf{w})} p_{j,y} (1 - 2y_i), & \forall i \in [n], \forall j = 1, \dots, K-1, \\ \sum_{x \in C_{S_j}(\mathbf{w})} p_{j,x} &= 1, \quad p_{j,x} \geq 0, & \forall j, \forall x \in C_{S_j}(\mathbf{w}). \end{aligned} \quad (15)$$

We cast Eq. (15) as a standard linear feasibility program, $A_{\text{eq}} \mathbf{p} = b_{\text{eq}}$ with $\mathbf{p} \geq 0$, where $\mathbf{p}$ is the concatenated vector of all probabilities $\{p_{j,x}\}$. The constraint matrix A_{eq} has an integer block structure derived directly from Eq. (15): $(K-1)n$ rows enforce the matching of Z_i expectations, and K rows enforce the normalization for each logical state. For the $K=2$ case, this results in an $(n+2) \times (|C_{S_0}| + |C_{S_1}|)$ matrix. We solve this system numerically using standard LP solvers.

3.2.2 Rational reconstruction

Numerical LP solvers produce floating-point solutions, but we require exact rational solutions for rigorous verification and analytical insight. Although the LP matrix A_{eq} and right-hand side b_{eq} are integers, basic feasible solutions can have large denominators, and small numerical errors from floating-point arithmetic can cause the KL equalities to be violated upon simple rounding. The Search Agent therefore employs two distinct strategies to convert numerical solutions into exact rationals, exploiting the underlying integer structure of the LP.

Exact BFS reconstruction. If the solver returns a solution that is close to a basic feasible solution (BFS), at most $(K-1)n + K$ entries will be nonzero. We can identify a full-rank basis B of this size, form the integer submatrix $A_B \in \mathbb{Z}^{((K-1)n+K) \times ((K-1)n+K)}$, and solve the system over the rational numbers $\mathbb{Q}$:

$$A_B \mathbf{p}_B = b_{\text{eq}}, \quad \text{with } p_i = 0 \text{ for } i \notin B.$$

We then verify that the resulting rational vector $\mathbf{p}$ satisfies $A_{\text{eq}} \mathbf{p} = b_{\text{eq}}$ and $\mathbf{p} \geq 0$ exactly. This procedure is outlined in Alg. 1 [56, 57, 58].

Algorithm 1 Exact BFS

Require: integer matrix A_{eq} , integer vector b_{eq} , numerical solution $\mathbf{p}^{(\text{num})}$

- 1: Choose a basis B of size $(K-1)n + K$ such that A_B is full-rank.
 - 2: Solve $A_B \mathbf{p}_B = b_{\text{eq}}$ over $\mathbb{Q}$; set $p_i = 0$ for $i \notin B$.
 - 3: Assert that $A_{\text{eq}} \mathbf{p} = b_{\text{eq}}$ and $\mathbf{p} \geq 0$ hold exactly.
 - 4: **return** rational vector $\mathbf{p}$.
-

Algorithm 2 Rationalize by projection

Require: integer A_{eq} , integer b_{eq} , numerical $\mathbf{p}^{(\text{num})}$, denominator bound D

- 1: $\tilde{p}_i \leftarrow \text{CFround}(p_i^{(\text{num})}; \text{den} \leq D)$ for all i .
 - 2: Solve $A_{\text{eq}} \mathbf{d} = b_{\text{eq}} - A_{\text{eq}} \tilde{\mathbf{p}}$ over $\mathbb{Q}$; set $\mathbf{p} \leftarrow \tilde{\mathbf{p}} + \mathbf{d}$.
 - 3: Enforce $\mathbf{p} \geq 0$ (clip if needed) and re-project exactly if clipping occurred.
 - 4: Renormalize each block; **return** rational vector $\mathbf{p}$.
-

Continued fractions + exact projection. When the solution is not clearly a BFS, we first rationalize each entry individually and then project the result back onto the affine subspace defined by the constraints. We define

$$\tilde{p}_i = \text{CFround}(p_i^{(\text{num})}; \text{den} \leq D) \quad (i = 1, \dots, N),$$

where CFround returns the nearest rational number with a denominator no larger than D ; $N = \sum_{j=0}^{K-1} |C_{S_j}|$. Next, we compute a rational correction vector $\mathbf{d} \in \mathbb{Q}^N$ by solving

$$A_{\text{eq}}(\tilde{\mathbf{p}} + \mathbf{d}) = b_{\text{eq}} \iff A_{\text{eq}} \mathbf{d} = b_{\text{eq}} - A_{\text{eq}} \tilde{\mathbf{p}}$$

exactly over $\mathbb{Q}$, and set $\mathbf{p} \leftarrow \tilde{\mathbf{p}} + \mathbf{d}$. If small negative entries arise due to rounding, we clip them to 0 and re-project the vector exactly to restore $A_{\text{eq}} \mathbf{p} = b_{\text{eq}}$. Finally, we renormalize each block so that $\sum_{u \in C_{S_j}} p_{j,u} = 1$. This procedure, summarized in Alg. 2, relies on the best-approximation properties of continued fractions combined with exact rational linear algebra [59, 60, 61].

3.2.3 Outputs and algorithmic skeleton

Reported quantities. For each successful hit, we record the parameters $(n, m, K, \mathbf{w}, \mathbf{S})$, the rational probabilities $\{p_{j,x}\}$, the explicit logical states $\{|j_L\rangle\}$, the KL expectation values $\langle Z_i \rangle$, and the logical order $\mathcal{O}$.

Skeleton (general K). The complete search procedure is summarized in Alg. 3.

3.3 Audit agent

The Search Agent produces candidate codes with claimed properties. To ensure correctness, we maintain an independent verification pipeline managed by the Audit Agent as presented in Alg. 4. To guarantee the reliability of the results, the Audit Agent independently generates a code checker based on the problem definition. In addition, audit agent also use the long-reasoning workflow agent from TeXRA to conduct the analytical checks. In addition to this automated audit, the researcher performs an independent manual inspection.

Verification of the KL condition. With the rational probabilities $\{p_{j,x}\}$, the logical states are formed as $|j_L\rangle = \sum_{x \in C_{S_j}(\mathbf{w})} \sqrt{p_{j,x}} |x\rangle$. The audit verifies two conditions for each site i and each Pauli operator $P_i \in \{X_i, Y_i, Z_i\}$: (i) equal diagonal elements, $\langle j_L | P_i | j_L \rangle = \langle k_L | P_i | k_L \rangle$ for all j, k ; and (ii) vanishing off-diagonal elements, $\langle j_L | P_i | k_L \rangle = 0$ for all $j \neq k$. These checks are performed using both floating-point and exact rational arithmetic against a specified tolerance.

Verification of the transversal gate. The audit confirms that the transversal operator $U(\mathbf{w}, m)$ acts as an eigenoperator on each logical state. This is done by applying $U(\mathbf{w}, m)$ to $|j_L\rangle$ and then projecting the resulting state back onto the original via the inner product $\langle j_L | U(\mathbf{w}, m) | j_L \rangle$. This projection must yield the expected eigenvalue $\omega_m^{S_j}$ and confirm that the state is indeed an eigenvector.

Algorithm 3 Search for subset-sum quantum codes (general K , distance = 2)

Require: $n, m, K \geq 2$

```

1:  $\mathcal{HITS} \leftarrow \emptyset$ 
2: for all nondecreasing  $\mathbf{w} \in \{1, \dots, m-1\}^n$  do
3:    $\mathbf{S} \leftarrow \{(S_1, \dots, S_{K-1}) : 1 \leq S_1 < \dots < S_{K-1} \leq m-1, \text{ satisfying screen (13)}\}$ 
4:   for all  $(S_1, \dots, S_{K-1}) \in \mathbf{S}$  do
5:     Build  $C_{S_j}(\mathbf{w})$  for  $j = 0, \dots, K-1$  (with  $S_0 = 0$ ); continue if any  $C_{S_j}(\mathbf{w}) = \emptyset$ .
6:     Compute  $d(C)$  for  $C := \bigcup_{j=0}^{K-1} C_{S_j}(\mathbf{w})$ ; continue if  $d(C) \neq 2$ .
7:     Solve LP (15) for  $\{p_{j,x}\}$ ; continue if infeasible.
8:     Perform rational reconstruction (Alg. 1 or 2).
9:     Assemble  $|j_L\rangle$ ; verify KL conditions for  $E \in \{X, Y, Z\}$ ; continue if any check fails.
10:    Verify  $U(\mathbf{w}, m) |j_L\rangle \propto \omega_m^{S_j} |j_L\rangle$ ; continue if this fails.
11:    Append the full record to  $\mathcal{HITS}$ .
12:   end for
13: end for
14: return  $\mathcal{HITS}$ 

```

Algorithm 4 Audit of search results (agent & researcher, independently)

Require: A record $R = (n, m, \mathbf{w}, \mathbf{S}, \{C_{S_j}(\mathbf{w})\}_{j=0}^{K-1}, \{p_{j,x}\})$, with $\sum_{x \in C_{S_j}} p_{j,x} = 1$.

1: Define tolerances: $\tau_{\text{float}} := 10^{-10}$, $\tau_{\text{rat}} := 0$.

Helper $\text{VERIFY}(R; \text{mode})$:

2: Build $|j_L\rangle = \sum_{x \in C_{S_j}(\mathbf{w})} \sqrt{p_{j,x}} |x\rangle$ for $j = 0, \dots, K-1$. Set τ based on mode.

3: **KL (Eq. 1):** Assert that for all i and all $P_i \in \{X_i, Y_i, Z_i\}$:

$$\begin{cases} |\langle j_L | P_i | j'_L \rangle| \leq \tau, & \forall j \neq j', \\ |\langle 0_L | P_i | 0_L \rangle - \langle j_L | P_i | j_L \rangle| \leq \tau, & \forall j. \end{cases}$$

4: **Transversal (Eq. 5):** Assert that for all j :

$$\|U(\mathbf{w}, m) |j_L\rangle - \omega_m^{S_j} |j_L\rangle\| \leq \tau.$$

5: Return PASS if and only if all assertions hold.

Agent audit: Run $\text{VERIFY}(R; \text{float})$ and $\text{VERIFY}(R; \text{rational})$.

Human audit: Run $\text{VERIFY}(R; \text{rational})$.

6: **Accept** R if all three calls pass; otherwise, **Flag** R .

The Audit Agent is prompted with the verification rules and independently generates a Python script to check the search results. This checker audits whether the found codes fulfill both the KL conditions and the transversality property. For all new quantum codes found for $n \in \{4, 5, 6\}$, the agent's audit passed. Additionally, a reasoning agent using the derivation-then-edit mode was used to check the analytical results presented in LaTeX form.

In addition to the agent audit, manual validation by the researcher is essential. This audit consists of two parts: first, we rerun the verification with independently written scripts on all search results; second, the KL and transversal conditions for all instances are verified manually.

The whole automatic SSLP workflow can be achieved by combining all the synthesis, search, and audit agents together. Through a complete sweep—parameters, rational certificates, full logical-state expansions, and induced diagonal-transversal orders $\mathcal{O}$, a large number of quantum error-correction codes can be found by the multi-agent orchestration as presented in the next section.

4 Search results

Executing the pipeline described in Section 3, our multi-agent system performed large-scale parameter sweeps on a local high-performance computing workstation equipped with an Intel Xeon w7-3565X CPU and two Nvidia RTX 6000 Ada Generation GPUs. The search focused on distance-2 codes for $n \leq 6$ qubits and logical dimensions $K \in \{2, 3, 4\}$. This procedure, executed by the Search Agent and verified

by the Audit Agent, yielded a rich catalog of new nonadditive quantum codes. The following subsections present these codes, organized by logical dimension K . Each entry includes the code parameters, the explicit logical states with exact rational amplitudes, and the resulting transversal diagonal gate.

4.1 $K = 2$ (two-dimensional logical space)

For two-dimensional codes ($K = 2$), our search on up to $n = 6$ qubits revealed a rich structure, including codes with cyclic group orders as high as 18. The following table summarizes the parameters of these instances, followed by their explicit logical state constructions.

Parameter table. One row per discovered instance (sorted by order).

Order $\mathcal{O}$	m	n	$\mathbf{w}$ (sorted)	$\mathbf{S} = (0, S_1)$
2	4	4	(1,1,1,1)	(0,2)
3	6	5	(1,1,1,1,3)	(0,4)
4	8	5	(1,1,1,3,3)	(0,6)
5	10	5	(1,1,4,4,4)	(0,2)
6	12	5	(4,4,4,6,6)	(0,2)
7	14	5	(2,2,2,4,4)	(0,6)
8	16	5	(2,2,4,4,8)	(0,10)
9	18	5	(2,2,4,4,6)	(0,8)
10	10	6	(1,1,1,1,4,6)	(0,7)
11	11	6	(1,1,1,1,4,4)	(0,8)
12	12	6	(1,1,1,2,3,4)	(0,5)
13	13	6	(1,1,1,2,5,5)	(0,10)
14	14	6	(1,1,1,3,3,6)	(0,9)
15	15	6	(1,1,2,2,5,6)	(0,11)
16	16	6	(1,1,2,3,4,5)	(0,7)
17	17	6	(1,1,2,4,4,6)	(0,8)
18	18	6	(1,2,3,4,5,6)	(0,11)

Order 2 ($m = 4$, $\mathbf{w} = (1, 1, 1, 1)$, $\mathbf{S} = (0, 2)$; sizes $(|C_0|, |C_2|) = (2, 6)$).

$$U(\mathbf{w}, 4) = Z\left(\frac{2\pi}{4}\right)^{\otimes 4}, \quad \bar{U} = \text{diag}(1, \omega_4^2).$$

$$\begin{aligned} |0_L\rangle &= \frac{1}{\sqrt{2}} |0000\rangle + \frac{1}{\sqrt{2}} |1111\rangle, \\ |1_L\rangle &= \frac{1}{\sqrt{2}} |0011\rangle + \frac{1}{\sqrt{2}} |1100\rangle. \end{aligned}$$

Order 3 ($m = 6$, $\mathbf{w} = (1, 1, 1, 1, 3)$, $\mathbf{S} = (0, 4)$; sizes $(5, 5)$).

$$U(\mathbf{w}, 6) = Z\left(\frac{2\pi}{6}\right)^{\otimes 4} \otimes Z\left(\frac{6\pi}{6}\right), \quad \bar{U} = \text{diag}(1, \omega_6^4).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{1}{3}} |00000\rangle + \sqrt{\frac{1}{3}} |01111\rangle + \sqrt{\frac{1}{3}} |10111\rangle, \\ |1_L\rangle &= \sqrt{\frac{1}{3}} |00011\rangle + \sqrt{\frac{1}{3}} |00101\rangle + \sqrt{\frac{1}{3}} |11110\rangle. \end{aligned}$$

Order 4 ($m = 8$, $\mathbf{w} = (1, 1, 1, 3, 3)$, $\mathbf{S} = (0, 6)$; sizes $(4, 3)$).

$$U(\mathbf{w}, 8) = Z\left(\frac{2\pi}{8}\right)^{\otimes 3} \otimes Z\left(\frac{6\pi}{8}\right)^{\otimes 2}, \quad \bar{U} = \text{diag}(1, \omega_8^6).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{1}{4}} |00000\rangle + \sqrt{\frac{1}{4}} |01111\rangle + \sqrt{\frac{1}{4}} |10111\rangle + \sqrt{\frac{1}{4}} |11011\rangle, \\ |1_L\rangle &= \sqrt{\frac{1}{2}} |00011\rangle + \sqrt{\frac{1}{4}} |11101\rangle + \sqrt{\frac{1}{4}} |11110\rangle. \end{aligned}$$

Order 5 ($m = 10$, $\mathbf{w} = (1, 1, 4, 4, 4)$, $\mathbf{S} = (0, 2)$; sizes $(4, 2)$).

$$U(\mathbf{w}, 10) = Z\left(\frac{2\pi}{10}\right)^{\otimes 2} \otimes Z\left(\frac{8\pi}{10}\right)^{\otimes 3}, \quad \bar{U} = \text{diag}(1, \omega_{10}^2).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{2}{5}} |00000\rangle + \sqrt{\frac{1}{5}} |11011\rangle + \sqrt{\frac{1}{5}} |11101\rangle + \sqrt{\frac{1}{5}} |11110\rangle, \\ |1_L\rangle &= \sqrt{\frac{2}{5}} |00111\rangle + \sqrt{\frac{3}{5}} |11000\rangle. \end{aligned}$$

Order 6 ($m = 12$, $\mathbf{w} = (4, 4, 4, 6, 6)$, $\mathbf{S} = (0, 2)$; sizes $(4, 6)$).

$$U(\mathbf{w}, 12) = Z\left(\frac{8\pi}{12}\right)^{\otimes 3} \otimes Z\left(\frac{12\pi}{12}\right)^{\otimes 2}, \quad \bar{U} = \text{diag}(1, \omega_{12}^2).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{1}{3}} |00011\rangle + \sqrt{\frac{1}{2}} |11100\rangle + \sqrt{\frac{1}{6}} |11111\rangle, \\ |1_L\rangle &= \sqrt{\frac{1}{3}} |01110\rangle + \sqrt{\frac{1}{3}} |10101\rangle + \sqrt{\frac{1}{6}} |11001\rangle + \sqrt{\frac{1}{6}} |11010\rangle. \end{aligned}$$

Order 7 ($m = 14$, $\mathbf{w} = (2, 2, 2, 4, 4)$, $\mathbf{S} = (0, 6)$; sizes $(2, 7)$).

$$U(\mathbf{w}, 14) = Z\left(\frac{4\pi}{14}\right)^{\otimes 3} \otimes Z\left(\frac{8\pi}{14}\right)^{\otimes 2}, \quad \bar{U} = \text{diag}(1, \omega_{14}^6).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{4}{7}} |00000\rangle + \sqrt{\frac{3}{7}} |11111\rangle, \\ |1_L\rangle &= \sqrt{\frac{2}{7}} |00101\rangle + \sqrt{\frac{2}{7}} |01010\rangle + \sqrt{\frac{1}{7}} |10001\rangle + \sqrt{\frac{1}{7}} |10010\rangle + \sqrt{\frac{1}{7}} |11100\rangle. \end{aligned}$$

Order 8 ($m = 16$, $\mathbf{w} = (2, 2, 4, 4, 8)$, $\mathbf{S} = (0, 10)$; sizes $(4, 4)$).

$$U(\mathbf{w}, 16) = Z\left(\frac{4\pi}{16}\right)^{\otimes 2} \otimes Z\left(\frac{8\pi}{16}\right)^{\otimes 2} \otimes Z\left(\frac{16\pi}{16}\right), \quad \bar{U} = \text{diag}(1, \omega_{16}^{10}).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{3}{8}} |00000\rangle + \sqrt{\frac{1}{8}} |00111\rangle + \sqrt{\frac{1}{4}} |11011\rangle + \sqrt{\frac{1}{4}} |11101\rangle, \\ |1_L\rangle &= \sqrt{\frac{1}{8}} |01001\rangle + \sqrt{\frac{3}{8}} |01110\rangle + \sqrt{\frac{1}{2}} |10001\rangle. \end{aligned}$$

Order 9 ($m = 18$, $\mathbf{w} = (2, 2, 4, 4, 6)$, $\mathbf{S} = (0, 8)$; sizes $(2, 5)$).

$$U(\mathbf{w}, 18) = Z\left(\frac{4\pi}{18}\right)^{\otimes 2} \otimes Z\left(\frac{8\pi}{18}\right)^{\otimes 2} \otimes Z\left(\frac{12\pi}{18}\right), \quad \bar{U} = \text{diag}(1, \omega_{18}^8).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{5}{9}} |00000\rangle + \sqrt{\frac{4}{9}} |11111\rangle, \\ |1_L\rangle &= \sqrt{\frac{1}{3}} |00110\rangle + \sqrt{\frac{2}{9}} |01001\rangle + \sqrt{\frac{2}{9}} |10001\rangle + \sqrt{\frac{1}{9}} |11010\rangle + \sqrt{\frac{1}{9}} |11100\rangle. \end{aligned}$$

Order 10 ($m = 10$, $\mathbf{w} = (1, 1, 1, 1, 4, 6)$, $\mathbf{S} = (0, 7)$; sizes $(3, 8)$).

$$U(\mathbf{w}, 10) = Z\left(\frac{2\pi}{10}\right)^{\otimes 4} \otimes Z\left(\frac{8\pi}{10}\right) \otimes Z\left(\frac{12\pi}{10}\right), \quad \bar{U} = \text{diag}(1, \omega_{10}^7).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{3}{10}} |000000\rangle + \sqrt{\frac{3}{10}} |000011\rangle + \sqrt{\frac{2}{5}} |111101\rangle, \\ |1_L\rangle &= \sqrt{\frac{1}{10}} |000101\rangle + \sqrt{\frac{1}{10}} |001001\rangle + \sqrt{\frac{2}{5}} |010001\rangle + \sqrt{\frac{1}{10}} |100001\rangle + \sqrt{\frac{3}{10}} |101110\rangle. \end{aligned}$$

Order 11 ($m = 11$, $\mathbf{w} = (1, 1, 1, 1, 4, 4)$, $\mathbf{S} = (0, 8)$; sizes $(5, 3)$).

$$U(\mathbf{w}, 11) = Z\left(\frac{2\pi}{11}\right)^{\otimes 4} \otimes Z\left(\frac{8\pi}{11}\right)^{\otimes 2}, \quad \bar{U} = \text{diag}(1, \omega_{11}^8).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{3}{11}} |000000\rangle + \sqrt{\frac{2}{11}} |011111\rangle + \sqrt{\frac{2}{11}} |101111\rangle + \sqrt{\frac{2}{11}} |110111\rangle + \sqrt{\frac{2}{11}} |111011\rangle, \\ |1_L\rangle &= \sqrt{\frac{5}{11}} |000011\rangle + \sqrt{\frac{3}{11}} |111101\rangle + \sqrt{\frac{3}{11}} |111110\rangle. \end{aligned}$$

Order 12 ($m = 12$, $\mathbf{w} = (1, 1, 1, 2, 3, 4)$, $\mathbf{S} = (0, 5)$; sizes $(2, 8)$).

$$U(\mathbf{w}, 12) = Z\left(\frac{2\pi}{12}\right)^{\otimes 3} \otimes Z\left(\frac{4\pi}{12}\right) \otimes Z\left(\frac{6\pi}{12}\right) \otimes Z\left(\frac{8\pi}{12}\right), \quad \bar{U} = \text{diag}(1, \omega_{12}^5).$$

$$|0_L\rangle = \sqrt{\frac{7}{12}} |000000\rangle + \sqrt{\frac{5}{12}} |111111\rangle,$$

$$|1_L\rangle = \sqrt{\frac{1}{4}} |000110\rangle + \sqrt{\frac{1}{12}} |001001\rangle + \sqrt{\frac{1}{4}} |010001\rangle + \sqrt{\frac{1}{12}} |100001\rangle + \sqrt{\frac{1}{6}} |101010\rangle + \sqrt{\frac{1}{6}} |111100\rangle.$$

Order 13 ($m = 13$, $\mathbf{w} = (1, 1, 1, 2, 5, 5)$, $\mathbf{S} = (0, 10)$; sizes $(5, 3)$).

$$U(\mathbf{w}, 13) = Z\left(\frac{2\pi}{13}\right)^{\otimes 3} \otimes Z\left(\frac{4\pi}{13}\right) \otimes Z\left(\frac{10\pi}{13}\right)^{\otimes 2}, \quad \bar{U} = \text{diag}(1, \omega_{13}^{10}).$$

$$|0_L\rangle = \sqrt{\frac{3}{13}} |000000\rangle + \sqrt{\frac{2}{13}} |001111\rangle + \sqrt{\frac{2}{13}} |010111\rangle + \sqrt{\frac{2}{13}} |100111\rangle + \sqrt{\frac{4}{13}} |111011\rangle,$$

$$|1_L\rangle = \sqrt{\frac{7}{13}} |000011\rangle + \sqrt{\frac{3}{13}} |111101\rangle + \sqrt{\frac{3}{13}} |111110\rangle.$$

Order 14 ($m = 14$, $\mathbf{w} = (1, 1, 1, 3, 3, 6)$, $\mathbf{S} = (0, 9)$; sizes $(4, 4)$).

$$U(\mathbf{w}, 14) = Z\left(\frac{2\pi}{14}\right)^{\otimes 3} \otimes Z\left(\frac{6\pi}{14}\right)^{\otimes 2} \otimes Z\left(\frac{12\pi}{14}\right), \quad \bar{U} = \text{diag}(1, \omega_{14}^9).$$

$$|0_L\rangle = \sqrt{\frac{5}{14}} |000000\rangle + \sqrt{\frac{3}{14}} |011111\rangle + \sqrt{\frac{3}{14}} |101111\rangle + \sqrt{\frac{3}{14}} |110111\rangle,$$

$$|1_L\rangle = \sqrt{\frac{2}{7}} |000011\rangle + \sqrt{\frac{2}{7}} |000101\rangle + \sqrt{\frac{1}{14}} |111001\rangle + \sqrt{\frac{5}{14}} |111110\rangle.$$

Order 15 ($m = 15$, $\mathbf{w} = (1, 1, 2, 2, 5, 6)$, $\mathbf{S} = (0, 11)$; sizes $(4, 4)$).

$$U(\mathbf{w}, 15) = Z\left(\frac{2\pi}{15}\right)^{\otimes 2} \otimes Z\left(\frac{4\pi}{15}\right)^{\otimes 2} \otimes Z\left(\frac{10\pi}{15}\right) \otimes Z\left(\frac{12\pi}{15}\right), \quad \bar{U} = \text{diag}(1, \omega_{15}^{11}).$$

$$|0_L\rangle = \sqrt{\frac{4}{15}} |000000\rangle + \sqrt{\frac{1}{3}} |001111\rangle + \sqrt{\frac{1}{5}} |110111\rangle + \sqrt{\frac{1}{5}} |111011\rangle,$$

$$|1_L\rangle = \sqrt{\frac{7}{15}} |000011\rangle + \sqrt{\frac{2}{15}} |011101\rangle + \sqrt{\frac{2}{15}} |101101\rangle + \sqrt{\frac{4}{15}} |111110\rangle.$$

Order 16 ($m = 16$, $\mathbf{w} = (1, 1, 2, 3, 4, 5)$, $\mathbf{S} = (0, 7)$; sizes $(2, 6)$).

$$U(\mathbf{w}, 16) = Z\left(\frac{2\pi}{16}\right)^{\otimes 2} \otimes Z\left(\frac{4\pi}{16}\right) \otimes Z\left(\frac{6\pi}{16}\right) \otimes Z\left(\frac{8\pi}{16}\right) \otimes Z\left(\frac{10\pi}{16}\right), \quad \bar{U} = \text{diag}(1, \omega_{16}^7).$$

$$|0_L\rangle = \sqrt{\frac{9}{16}} |000000\rangle + \sqrt{\frac{7}{16}} |111111\rangle,$$

$$|1_L\rangle = \sqrt{\frac{5}{16}} |000110\rangle + \sqrt{\frac{3}{16}} |001001\rangle + \sqrt{\frac{1}{16}} |011010\rangle + \sqrt{\frac{1}{16}} |101010\rangle + \sqrt{\frac{1}{4}} |110001\rangle + \sqrt{\frac{1}{8}} |111100\rangle.$$

Order 17 ($m = 17$, $\mathbf{w} = (1, 1, 2, 4, 4, 6)$, $\mathbf{S} = (0, 8)$; sizes $(3, 5)$).

$$U(\mathbf{w}, 17) = Z\left(\frac{2\pi}{17}\right)^{\otimes 2} \otimes Z\left(\frac{4\pi}{17}\right) \otimes Z\left(\frac{8\pi}{17}\right)^{\otimes 2} \otimes Z\left(\frac{12\pi}{17}\right), \quad \bar{U} = \text{diag}(1, \omega_{17}^8).$$

$$|0_L\rangle = \sqrt{\frac{9}{17}} |000000\rangle + \sqrt{\frac{4}{17}} |011111\rangle + \sqrt{\frac{4}{17}} |101111\rangle,$$

$$|1_L\rangle = \sqrt{\frac{7}{17}} |000110\rangle + \sqrt{\frac{6}{17}} |001001\rangle + \sqrt{\frac{2}{17}} |110001\rangle + \sqrt{\frac{1}{17}} |111010\rangle + \sqrt{\frac{1}{17}} |111100\rangle.$$

Order 18 ($m = 18$, $\mathbf{w} = (1, 2, 3, 4, 5, 6)$, $\mathbf{S} = (0, 11)$; sizes $(3, 5)$).

$$U(\mathbf{w}, 18) = \bigotimes_{i=1}^6 Z\left(\frac{2\pi i}{18}\right), \quad \bar{U} = \text{diag}(1, \omega_{18}^{11}).$$

$$|0_L\rangle = \sqrt{\frac{7}{18}} |000000\rangle + \sqrt{\frac{1}{6}} |001111\rangle + \sqrt{\frac{4}{9}} |110111\rangle,$$

$$|1_L\rangle = \sqrt{\frac{2}{9}} |000011\rangle + \sqrt{\frac{5}{18}} |010110\rangle + \sqrt{\frac{1}{18}} |011001\rangle + \sqrt{\frac{1}{3}} |100101\rangle + \sqrt{\frac{1}{9}} |111010\rangle.$$

4.2 $K = 3$ (qutrit logical space)

For $K = 3$ on $n = 6$ qubits, the search yielded codes with orders up to 16. The attainable orders appear less continuous in m compared to the $K = 2$ case. Key instances are detailed below.

Parameter table.

Order $\mathcal{O}$	m	n	$\mathbf{w}$ (sorted)	$\mathbf{S} = (0, S_1, S_2)$
3	6	6	(1,1,1,1,3,3)	(0,2,4)
4	8	6	(1,1,1,3,3,3)	(0,2,4)
6	12	6	(1,1,1,5,5,7)	(0,6,10)
8	16	6	(1,1,4,4,7,7)	(0,2,8)
10	10	6	(1,1,1,4,4,4)	(0,2,5)
12	12	6	(2,2,3,3,4,4)	(0,6,7)
14	14	6	(1,1,3,4,6,6)	(0,2,7)
15	15	6	(1,1,4,4,6,9)	(0,2,10)
16	16	6	(1,2,4,4,6,7)	(0,8,11)

Logical states:

Order 3 ($m = 6$, $\mathbf{w} = (1, 1, 1, 1, 3, 3)$, $\mathbf{S} = (0, 2, 4)$).

$$U = Z\left(\frac{2\pi}{6}\right)^{\otimes 4} \otimes Z\left(\frac{6\pi}{6}\right)^{\otimes 2}, \quad \bar{U} = \text{diag}(1, \omega_6^2, \omega_6^4).$$

$$|0_L\rangle = \sqrt{\frac{1}{3}}|000000\rangle + \sqrt{\frac{1}{3}}|011110\rangle + \sqrt{\frac{1}{3}}|101101\rangle,$$

$$|1_L\rangle = \sqrt{\frac{1}{3}}|011000\rangle + \sqrt{\frac{1}{3}}|001100\rangle + \sqrt{\frac{1}{3}}|100111\rangle,$$

$$|2_L\rangle = \sqrt{\frac{1}{3}}|111100\rangle + \sqrt{\frac{1}{3}}|001010\rangle + \sqrt{\frac{1}{3}}|000101\rangle.$$

Order 4 ($m = 8$, $\mathbf{w} = (1, 1, 1, 3, 3, 3)$, $\mathbf{S} = (0, 2, 4)$).

$$U = Z\left(\frac{2\pi}{8}\right)^{\otimes 3} \otimes Z\left(\frac{6\pi}{8}\right)^{\otimes 3}, \quad \bar{U} = \text{diag}(1, \omega_8^2, \omega_8^4).$$

$$|0_L\rangle = \sqrt{\frac{1}{4}}|000000\rangle + \sqrt{\frac{1}{4}}|110110\rangle + \sqrt{\frac{1}{4}}|101101\rangle + \sqrt{\frac{1}{4}}|101011\rangle,$$

$$|1_L\rangle = \sqrt{\frac{1}{4}}|101000\rangle + \sqrt{\frac{1}{4}}|011000\rangle + \sqrt{\frac{1}{2}}|100111\rangle,$$

$$|2_L\rangle = \sqrt{\frac{1}{4}}|100100\rangle + \sqrt{\frac{1}{4}}|001010\rangle + \sqrt{\frac{1}{4}}|100001\rangle + \sqrt{\frac{1}{4}}|111111\rangle.$$

Order 6 ($m = 12$, $\mathbf{w} = (1, 1, 1, 5, 5, 7)$, $\mathbf{S} = (0, 6, 10)$).

$$U = Z\left(\frac{2\pi}{12}\right)^{\otimes 3} \otimes Z\left(\frac{10\pi}{12}\right)^{\otimes 2} \otimes Z\left(\frac{14\pi}{12}\right), \quad \bar{U} = \text{diag}(1, \omega_{12}^6, \omega_{12}^{10}).$$

$$|0_L\rangle = \sqrt{\frac{1}{6}}|000000\rangle + \sqrt{\frac{1}{6}}|110110\rangle + \sqrt{\frac{1}{6}}|101110\rangle + \sqrt{\frac{1}{6}}|011110\rangle + \sqrt{\frac{1}{6}}|000101\rangle + \sqrt{\frac{1}{6}}|000011\rangle,$$

$$|1_L\rangle = \sqrt{\frac{1}{3}}|001100\rangle + \sqrt{\frac{1}{3}}|100010\rangle + \sqrt{\frac{1}{3}}|010111\rangle,$$

$$|2_L\rangle = \sqrt{\frac{2}{3}}|000110\rangle + \sqrt{\frac{1}{3}}|111001\rangle.$$

Order 8 ($m = 16$, $\mathbf{w} = (1, 1, 4, 4, 7, 7)$, $\mathbf{S} = (0, 2, 8)$).

$$U = Z\left(\frac{2\pi}{16}\right)^{\otimes 2} \otimes Z\left(\frac{8\pi}{16}\right)^{\otimes 2} \otimes Z\left(\frac{14\pi}{16}\right)^{\otimes 2}, \quad \bar{U} = \text{diag}(1, \omega_{16}^2, \omega_{16}^8).$$

$$|0_L\rangle = \sqrt{\frac{3}{8}}|000000\rangle + \sqrt{\frac{1}{8}}|011110\rangle + \sqrt{\frac{1}{8}}|101101\rangle + \sqrt{\frac{3}{8}}|110011\rangle,$$

$$|1_L\rangle = \sqrt{\frac{1}{4}}|000111\rangle + \sqrt{\frac{1}{4}}|001011\rangle + \sqrt{\frac{1}{2}}|110000\rangle,$$

$$|2_L\rangle = \sqrt{\frac{1}{8}}|001100\rangle + \sqrt{\frac{3}{8}}|010010\rangle + \sqrt{\frac{3}{8}}|100001\rangle + \sqrt{\frac{1}{8}}|111111\rangle.$$

Order 10 ($m = 10$, $\mathbf{w} = (1, 1, 1, 4, 4, 4)$, $\mathbf{S} = (0, 2, 5)$).

$$U = Z\left(\frac{2\pi}{10}\right)^{\otimes 3} \otimes Z\left(\frac{8\pi}{10}\right)^{\otimes 3}, \quad \bar{U} = \text{diag}(1, \omega_{10}^2, \omega_{10}^5).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{2}{5}} |000000\rangle + \sqrt{\frac{1}{5}} |011110\rangle + \sqrt{\frac{1}{5}} |110101\rangle + \sqrt{\frac{1}{5}} |101011\rangle, \\ |1_L\rangle &= \sqrt{\frac{1}{5}} |110000\rangle + \sqrt{\frac{1}{5}} |101000\rangle + \sqrt{\frac{1}{5}} |011000\rangle + \sqrt{\frac{2}{5}} |000111\rangle, \\ |2_L\rangle &= \sqrt{\frac{3}{10}} |010100\rangle + \sqrt{\frac{3}{10}} |001010\rangle + \sqrt{\frac{3}{10}} |100001\rangle + \sqrt{\frac{1}{10}} |111111\rangle. \end{aligned}$$

Order 12 ($m = 12$, $\mathbf{w} = (2, 2, 3, 3, 4, 4)$, $\mathbf{S} = (0, 6, 7)$).

$$U = Z\left(\frac{4\pi}{12}\right)^{\otimes 2} \otimes Z\left(\frac{6\pi}{12}\right)^{\otimes 2} \otimes Z\left(\frac{8\pi}{12}\right)^{\otimes 2}, \quad \bar{U} = \text{diag}(1, \omega_{12}^6, \omega_{12}^7).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{5}{12}} |000000\rangle + \sqrt{\frac{1}{4}} |011110\rangle + \sqrt{\frac{1}{4}} |101110\rangle + \sqrt{\frac{1}{12}} |110011\rangle, \\ |1_L\rangle &= \sqrt{\frac{5}{12}} |001100\rangle + \sqrt{\frac{1}{4}} |010010\rangle + \sqrt{\frac{1}{4}} |100010\rangle + \sqrt{\frac{1}{12}} |111111\rangle, \\ |2_L\rangle &= \sqrt{\frac{1}{2}} |000110\rangle + \sqrt{\frac{1}{12}} |001001\rangle + \sqrt{\frac{1}{12}} |001010\rangle + \sqrt{\frac{1}{3}} |111000\rangle. \end{aligned}$$

Order 14 ($m = 14$, $\mathbf{w} = (1, 1, 3, 4, 6, 6)$, $\mathbf{S} = (0, 2, 7)$).

$$U = Z\left(\frac{2\pi}{14}\right)^{\otimes 2} \otimes Z\left(\frac{6\pi}{14}\right) \otimes Z\left(\frac{8\pi}{14}\right) \otimes Z\left(\frac{12\pi}{14}\right)^{\otimes 2}, \quad \bar{U} = \text{diag}(1, \omega_{14}^2, \omega_{14}^7).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{2}{7}} |000000\rangle + \sqrt{\frac{1}{7}} |101110\rangle + \sqrt{\frac{1}{7}} |101101\rangle + \sqrt{\frac{3}{7}} |110011\rangle, \\ |1_L\rangle &= \sqrt{\frac{3}{7}} |110000\rangle + \sqrt{\frac{2}{7}} |101011\rangle + \sqrt{\frac{2}{7}} |000111\rangle, \\ |2_L\rangle &= \sqrt{\frac{1}{14}} |001100\rangle + \sqrt{\frac{1}{7}} |100010\rangle + \sqrt{\frac{3}{14}} |010010\rangle + \sqrt{\frac{5}{14}} |100001\rangle + \sqrt{\frac{3}{14}} |111111\rangle. \end{aligned}$$

Order 15 ($m = 15$, $\mathbf{w} = (1, 1, 4, 4, 6, 9)$, $\mathbf{S} = (0, 2, 10)$).

$$U = Z\left(\frac{2\pi}{15}\right)^{\otimes 2} \otimes Z\left(\frac{8\pi}{15}\right)^{\otimes 2} \otimes Z\left(\frac{12\pi}{15}\right) \otimes Z\left(\frac{18\pi}{15}\right), \quad \bar{U} = \text{diag}(1, \omega_{15}^2, \omega_{15}^{10}).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{1}{3}} |000011\rangle + \sqrt{\frac{1}{15}} |011110\rangle + \sqrt{\frac{1}{15}} |101110\rangle + \sqrt{\frac{4}{15}} |110101\rangle + \sqrt{\frac{4}{15}} |111001\rangle, \\ |1_L\rangle &= \sqrt{\frac{2}{5}} |001101\rangle + \sqrt{\frac{2}{15}} |110000\rangle + \sqrt{\frac{7}{15}} |110011\rangle, \\ |2_L\rangle &= \sqrt{\frac{1}{15}} |000110\rangle + \sqrt{\frac{1}{15}} |001010\rangle + \sqrt{\frac{4}{15}} |010001\rangle + \sqrt{\frac{4}{15}} |100001\rangle + \sqrt{\frac{1}{3}} |111111\rangle. \end{aligned}$$

Order 16 ($m = 16$, $\mathbf{w} = (1, 2, 4, 4, 6, 7)$, $\mathbf{S} = (0, 8, 11)$).

$$U = Z\left(\frac{2\pi}{16}\right) \otimes Z\left(\frac{4\pi}{16}\right) \otimes Z\left(\frac{8\pi}{16}\right)^{\otimes 2} \otimes Z\left(\frac{12\pi}{16}\right) \otimes Z\left(\frac{14\pi}{16}\right), \quad \bar{U} = \text{diag}(1, \omega_{16}^8, \omega_{16}^{11}).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{5}{16}} |000000\rangle + \sqrt{\frac{3}{16}} |011110\rangle + \sqrt{\frac{7}{16}} |101101\rangle + \sqrt{\frac{1}{16}} |110011\rangle, \\ |1_L\rangle &= \sqrt{\frac{7}{16}} |001100\rangle + \sqrt{\frac{1}{16}} |010010\rangle + \sqrt{\frac{5}{16}} |100001\rangle + \sqrt{\frac{3}{16}} |111111\rangle, \\ |2_L\rangle &= \sqrt{\frac{1}{8}} |000101\rangle + \sqrt{\frac{3}{8}} |001001\rangle + \sqrt{\frac{1}{4}} |100110\rangle + \sqrt{\frac{1}{4}} |111100\rangle. \end{aligned}$$

4.3 $K = 4$ (four-dimensional logical space)

For $K = 4$ on $n = 6$ qubits, our search identified fewer instances due to the increasing number of constraints. We report two notable codes with orders 4 and 6.

Parameter table.

Order $\mathcal{O}$	m	n	$\mathbf{w}$ (sorted)	$\mathbf{S} = (0, S_1, S_2, S_3)$
4	8	6	(1,1,1,3,3,3)	(0,2,4,6)
6	12	6	(1,1,3,3,5,5)	(0,2,6,10)

Logical states:

Order 4 ($m = 8, \mathbf{w} = (1, 1, 1, 3, 3, 3), \mathbf{S} = (0, 2, 4, 6)$).

$$U = Z\left(\frac{2\pi}{8}\right)^{\otimes 3} \otimes Z\left(\frac{6\pi}{8}\right)^{\otimes 3}, \quad \bar{U} = \text{diag}(1, \omega_8^2, \omega_8^4, \omega_8^6).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{1}{4}}|000000\rangle + \sqrt{\frac{1}{4}}|011110\rangle + \sqrt{\frac{1}{4}}|110101\rangle + \sqrt{\frac{1}{4}}|101011\rangle, \\ |1_L\rangle &= \sqrt{\frac{1}{2}}|110000\rangle + \sqrt{\frac{1}{2}}|001111\rangle, \\ |2_L\rangle &= \sqrt{\frac{1}{4}}|001100\rangle + \sqrt{\frac{1}{4}}|010010\rangle + \sqrt{\frac{1}{4}}|100001\rangle + \sqrt{\frac{1}{4}}|111111\rangle, \\ |3_L\rangle &= \sqrt{\frac{1}{2}}|000110\rangle + \sqrt{\frac{1}{2}}|111001\rangle. \end{aligned}$$

Order 6 ($m = 12, \mathbf{w} = (1, 1, 3, 3, 5, 5), \mathbf{S} = (0, 2, 6, 10)$).

$$U = Z\left(\frac{2\pi}{12}\right)^{\otimes 2} \otimes Z\left(\frac{6\pi}{12}\right)^{\otimes 2} \otimes Z\left(\frac{10\pi}{12}\right)^{\otimes 2}, \quad \bar{U} = \text{diag}(1, \omega_{12}^2, \omega_{12}^6, \omega_{12}^{10}).$$

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{1}{6}}|000000\rangle + \sqrt{\frac{1}{6}}|101110\rangle + \sqrt{\frac{1}{6}}|011101\rangle + \sqrt{\frac{1}{2}}|110011\rangle, \\ |1_L\rangle &= \sqrt{\frac{1}{3}}|110000\rangle + \sqrt{\frac{1}{3}}|101011\rangle + \sqrt{\frac{1}{3}}|010111\rangle, \\ |2_L\rangle &= \sqrt{\frac{1}{3}}|100010\rangle + \sqrt{\frac{1}{3}}|010001\rangle + \sqrt{\frac{1}{3}}|111111\rangle, \\ |3_L\rangle &= \sqrt{\frac{1}{3}}|111010\rangle + \sqrt{\frac{1}{3}}|110101\rangle + \sqrt{\frac{1}{3}}|000011\rangle. \end{aligned}$$

This systematic search produced a large number of new quantum codes for $n \in \{4, 5, 6\}$. The verified instances in this dataset reveal recurring patterns that can be elevated to analytical families.

5 From Instances to Analytical Families

The catalog of verified codes produced by the search is not an end in itself, but rather a dataset from which to extract deeper structure. The Synthesis Agent was tasked with analyzing these instances by human to identify and prove the existence of infinite, analytical families. Examining the verified instances in section 4, we identified several recurring structures that admit analytical generalization. In this section, we construct such distance-2 families analytically within the SSLP framework, using the same notation $(m, \mathbf{w}, \mathbf{S})$ and the same residue-shift separation rule that underpins the search.

5.1 $C_0 = \{0^n, 1^n\}$

Consider a code on $n \geq 2$ qubits with modulus $m \in \mathbb{Z}_{>0}$ and weight vector

$$\mathbf{w} = (1, 1, \dots, 1, m - (n - 1)) \in (\mathbb{Z}_m)^n.$$

We define the residue classes as

$$\begin{aligned} C_0 &= \{x \in \{0, 1\}^n : \langle \mathbf{w}, x \rangle \equiv 0 \pmod{m}\}, \\ C_s &= \{y \in \{0, 1\}^n : \langle \mathbf{w}, y \rangle \equiv s \pmod{m}\} \quad \text{for } s \in \{1, \dots, m-1\}. \end{aligned}$$

Realizing $C_0 = \{0^n, 1^n\}$. The residue class C_0 consists of only the two extremal strings, $C_0 = \{0^n, 1^n\}$, if and only if $m \geq n$. If $m < n$, a non-empty subsequence of the first $n - 1$ ones would sum to $0 \pmod{m}$.

Exact description of C_s (two slices). We can write any binary string y as (u, b) , where $u \in \{0, 1\}^{n-1}$ and $b \in \{0, 1\}$. Let $t = \text{wt}(u)$. The residue of y is $\langle \mathbf{w}, y \rangle \equiv t + b(m - (n - 1)) \pmod{m}$. This means that the class C_s is composed of two distinct Hamming-weight slices:

$$C_s = A_s \cup B_t, \quad \text{where } A_s = \{(u, 0) : \text{wt}(u) = s\}, \quad \text{and } B_t = \{(u, 1) : \text{wt}(u) = t\},$$

with $t = n - 1 + s - m$. Both slices are non-empty if and only if $m - (n - 1) \leq s \leq n - 1$.

Closed-form SSLP solution (exact Z -equalities). We choose $|0_L\rangle$ to be supported on C_0 and $|1_L\rangle$ on C_s . A site-symmetric solution that enforces $\langle Z_i \rangle_{|0_L\rangle} = \langle Z_i \rangle_{|1_L\rangle}$ for all i is given by the following probabilities: For $|0_L\rangle$, we set the probabilities on C_0 as

$$p(0^n) = 1 - \frac{s}{m}, \quad p(1^n) = \frac{s}{m}.$$

For $|1_L\rangle$, we set the probabilities on C_s as

$$q(y) = \begin{cases} \frac{m-s}{m} \frac{1}{\binom{n-1}{s}}, & \text{if } y \in A_s, \\ \frac{s}{m} \frac{1}{\binom{n-1}{t}}, & \text{if } y \in B_t, \quad \text{where } t = n - 1 + s - m. \end{cases}$$

This choice yields a common expectation value $\langle Z_i \rangle = 1 - \frac{2s}{m}$ for every site i .

Assembled code states. The resulting logical states are:

$$\begin{aligned} |0_L\rangle &= \sqrt{1 - \frac{s}{m}} |0^n\rangle + \sqrt{\frac{s}{m}} |1^n\rangle, \\ |1_L\rangle &= \sum_{y \in A_s} \sqrt{\frac{m-s}{m}} \frac{|y\rangle}{\sqrt{\binom{n-1}{s}}} + \sum_{y \in B_t} \sqrt{\frac{s}{m}} \frac{|y\rangle}{\sqrt{\binom{n-1}{t}}}, \quad \text{where } t = n - 1 + s - m. \end{aligned} \tag{16}$$

Transversal gate and logical action. The transversal gate $U(\mathbf{w}, m) = \bigotimes_{j=1}^n Z\left(\frac{2\pi w_j}{m}\right)$ acts on the code space as the logical diagonal gate $\bar{U} = \text{diag}(1, \omega_m^s)$, which has order $m/\text{gcd}(m, s)$.

Distance-2 condition (via the shift screen). If, in addition, the residue s satisfies

$$s \not\equiv \pm 1 \pmod{m} \quad \text{and} \quad s \not\equiv \pm(m - (n - 1)) \pmod{m},$$

then the residue-shift screen of Sec. 3.1 guarantees that all single-qubit X_i and Y_i off-diagonal terms vanish combinatorially. This ensures that the full weight-1 KL conditions hold, and the code has distance 2.

Explicit examples for $n = 5$. Here, $\mathbf{w} = (1, 1, 1, 1, m - 4)$, and the two-slice window for s is $m - 4 \leq s \leq 4$.

$m = 5, s = 2$. Here, $\mathbf{w} = (1, 1, 1, 1, 1)$ and $t = 1$. The logical states are:

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{3}{5}} |00000\rangle + \sqrt{\frac{2}{5}} |11111\rangle, \\ |1_L\rangle &= \frac{\sqrt{3/5}}{\sqrt{6}} \left(|11000\rangle + |10100\rangle + |10010\rangle + |01100\rangle + |01010\rangle + |00110\rangle \right) \\ &\quad + \frac{\sqrt{2/5}}{2} \left(|10001\rangle + |01001\rangle + |00101\rangle + |00011\rangle \right). \end{aligned}$$

The transversal gate $U(\mathbf{w}, 5) = Z(2\pi/5)^{\otimes 5}$ induces the logical gate $\bar{U} = \text{diag}(1, \omega_5^2)$.

Explicit examples for $n = 6$. Here, $\mathbf{w} = (1, 1, 1, 1, 1, m - 5)$. For $m = 7, \mathbf{w} = (1, 1, 1, 1, 1, 2)$, and the two-slice window is $2 \leq s \leq 5$. We consider $s = 3$, which satisfies the strict screen $s \not\equiv \pm 1, \pm 2 \pmod{7}$ and thus guarantees distance 2.

$m = 7, s = 3$. Here, $t = n-1+s-m = 1$.

$$\begin{aligned} |0_L\rangle &= \sqrt{\frac{4}{7}} |000000\rangle + \sqrt{\frac{3}{7}} |111111\rangle, \\ |1_L\rangle &= \frac{\sqrt{4/7}}{\sqrt{\binom{5}{3}}} \sum_{\text{wt}(u)=3} |u0\rangle + \frac{\sqrt{3/7}}{\sqrt{\binom{5}{1}}} \sum_{\text{wt}(u)=1} |u1\rangle. \end{aligned}$$

Expanding this in the computational basis (where the first five bits correspond to u and the last bit is written explicitly):

$$\begin{aligned} |1_L\rangle &= \frac{\sqrt{4/7}}{\sqrt{10}} \left(|111000\rangle + |110100\rangle + |110010\rangle + |101100\rangle + |101010\rangle \right. \\ &\quad \left. + |100110\rangle + |011100\rangle + |011010\rangle + |010110\rangle + |001110\rangle \right) \\ &\quad + \frac{\sqrt{3/7}}{\sqrt{5}} (|100001\rangle + |010001\rangle + |001001\rangle + |000101\rangle + |000011\rangle). \end{aligned}$$

The Z -equalities give $\langle Z_i \rangle = 1 - \frac{2s}{m} = 1/7$ for all sites. The transversal gate $U(\mathbf{w}, 7) = Z(2\pi/7)^{\otimes 5} \otimes Z(4\pi/7)$ induces the logical gate $\bar{U} = \text{diag}(1, \omega_7^3)$.

5.2 Family based on the even-parity subcode

We now construct a family of codes by restricting the supports to the even-parity subcode $\mathbf{E} = \{\sigma \in \{0, 1\}^n : \text{wt}(\sigma) \equiv 0 \pmod{2}\}$, where n is even.

Fix a modulus $m \geq 3$, a sorted weight vector $\mathbf{w} = (w_1, \dots, w_n) \in \{0, 1, \dots, m-1\}^n$, and a set of distinct residues $\mathbf{S} = \{S_0, \dots, S_{K-1}\} \subset \mathbb{Z}_m$ with $S_0 = 0$. We define the supports within the even subcode as

$$C_{S_k}^{(+)}(\mathbf{w}) = \left\{ \sigma \in \mathbf{E} : \sum_{j=1}^n w_j \sigma_j \equiv S_k \pmod{m} \right\},$$

and define the logical states as uniform superpositions over these supports:

$$|k_L\rangle = \frac{1}{\sqrt{|C_{S_k}^{(+)}|}} \sum_{\sigma \in C_{S_k}^{(+)}} |\sigma\rangle, \quad k = 0, \dots, K-1.$$

The physical diagonal transversal $U(\mathbf{w}, m) = \bigotimes_{j=1}^n Z(2\pi w_j/m)$ induces the logical gate

$$\bar{U} = \text{diag}(\omega_m^{S_0}, \dots, \omega_m^{S_{K-1}}).$$

Distance-2 Property. *Parity screen for X/Y errors.* Since each $|k_L\rangle$ is supported entirely on strings of even parity, a single-bit flip (an X_i or Y_i error) will map any basis state to a string of odd parity. Therefore, the resulting state has no overlap with any of the logical states, and all single-qubit X/Y KL conditions are automatically satisfied:

$$\langle k_L | X_i | k_L \rangle = \langle k_L | Y_i | k_L \rangle = \langle k_L | X_i | \ell_L \rangle = \langle k_L | Y_i | \ell_L \rangle = 0.$$

Column balance for Z errors. If each support set $C_{S_k}^{(+)}$ is “column-balanced”—meaning that for each bit position i , exactly half of the strings in the set have a ‘1’—then the expectation value $\langle k_L | Z_i | k_L \rangle$ will be zero for all k . Since the supports are disjoint, $\langle k_L | Z_i | \ell_L \rangle = 0$ for $k \neq \ell$. Thus, if column balance holds, all weight-1 KL conditions are satisfied, and the code has distance 2.

Sufficient symmetry for column balance. A convenient (but not necessary) way to ensure column balance is to choose $\mathbf{w}$ so that classes $C_{S_k}^{(+)}$ are closed under an involutive symmetry (e.g., bitwise complement when $\sum_j w_j \equiv 0 \pmod{m}$ and $S_k \equiv -S_k \pmod{m}$), or by introducing 0-weights / structured pairings that preserve residues). The examples below are constructed so that each listed residue class is column-balanced.

Family for general K . For any $K \geq 2$ and residue set $\mathbf{S} = \{0, S_1, \dots, S_{K-1}\}$, the logical diagonal gate $\bar{U}$ is cyclic with order

$$\mathcal{O} = \frac{m}{\gcd(m, S_1, \dots, S_{K-1})}.$$

For $K = 2$ and $\mathbf{S} = \{0, \Delta\}$, this simplifies to $\mathcal{O} = m/\gcd(\Delta, m)$.

5.2.1 Examples with $K = 2$

Example 1 ($n = 4, K = 2$; **order 2**). Take $m = 6$, $\mathbf{w} = (1, 2, 4, 5)$, and $\mathbf{S} = \{0, 3\}$. The supports are:

$$\begin{aligned} C_0^{(+)} &= \{0000, 0110, 1001, 1111\}, \\ C_3^{(+)} &= \{0011, 1100\}. \end{aligned}$$

The logical states are

$$|0_L\rangle = \frac{1}{2}(|0000\rangle + |0110\rangle + |1001\rangle + |1111\rangle), \quad |1_L\rangle = \frac{1}{\sqrt{2}}(|0011\rangle + |1100\rangle).$$

Both supports are column-balanced, so $\langle Z_i \rangle = 0$ for both states. The logical action is $\bar{U} = \text{diag}(1, e^{i\pi})$ (order 2).

Example 2 ($n = 6, K = 2$; **order 2**). Take $m = 8$, $\mathbf{w} = (1, 2, 3, 5, 6, 7)$, and $\mathbf{S} = \{0, 4\}$. The supports are:

$$\begin{aligned} C_0^{(+)} &= \{000000, 001100, 010010, 011110, 100001, 101101, 110011, 111111\}, \\ C_4^{(+)} &= \{000101, 010111, 101000, 111010\}. \end{aligned}$$

The logical states are uniform superpositions over these sets:

$$|0_L\rangle = \frac{1}{\sqrt{8}} \sum_{s \in C_0^{(+)}} |s\rangle, \quad |1_L\rangle = \frac{1}{2} \sum_{s \in C_4^{(+)}} |s\rangle.$$

Both supports are column-balanced, so $\langle Z_i \rangle = 0$. The logical action is $\bar{U} = \text{diag}(1, \omega_8^4) = \text{diag}(1, -1)$ (order 2).

Example 3 ($n = 6, K = 2$; **order 4**). Take $m = 8$, $\mathbf{w} = (6, 4, 0, 2, 7, 5)$, and $\mathbf{S} = \{0, 2\}$. The supports are:

$$\begin{aligned} C_0^{(+)} &= \{000000, 011011, 100100, 111111\}, \\ C_2^{(+)} &= \{001100, 010111, 101011, 110000\}. \end{aligned}$$

with the logic states:

$$|0_L\rangle = \frac{1}{2}(|000000\rangle + |011011\rangle + |100100\rangle + |111111\rangle), \quad |1_L\rangle = \frac{1}{2}(|001100\rangle + |010111\rangle + |101011\rangle + |110000\rangle).$$

Z_i check: one-counts are $[2, 2, 2, 2, 2, 2]$ on both $C_0^{(+)}$ and $C_2^{(+)}$ (each size 4); hence $\langle Z_i \rangle = 0$ for both. Logical action: $\bar{U} = \text{diag}(1, \omega_8^2)$ (order $8/\gcd(8, 2) = 4$).

5.2.2 Example with $K > 2$

Example 4 ($n = 6, K = 3$; **order 3**). Take $m = 9$, $\mathbf{w} = (1, 2, 5, 5, 7, 1)$, and $\mathbf{S} = \{0, 3, 6\}$. The supports are:

$$\begin{aligned} C_0^{(+)} &= \{000000, 001111, 010010, 101110, 110101, 111001\}, \\ C_3^{(+)} &= \{000110, 001010, 010001, 101101, 110000, 111111\}, \\ C_6^{(+)} &= \{000101, 001001, 010111, 011011, 100100, 101000, 110110, 111010\}. \end{aligned}$$

$$\begin{aligned} |0_L\rangle &= \frac{1}{\sqrt{6}}(|000000\rangle + |001111\rangle + |010010\rangle + |101110\rangle + |110101\rangle + |111001\rangle), \\ |1_L\rangle &= \frac{1}{\sqrt{6}}(|000110\rangle + |001010\rangle + |010001\rangle + |101101\rangle + |110000\rangle + |111111\rangle), \\ |2_L\rangle &= \frac{1}{\sqrt{8}}(|000101\rangle + |001001\rangle + |010111\rangle + |011011\rangle + |100100\rangle + |101000\rangle + |110110\rangle + |111010\rangle). \end{aligned}$$

Z_i check: one-counts are $[3, 3, 3, 3, 3, 3]$ on $C_0^{(+)}$ (size 6), $[3, 3, 3, 3, 3, 3]$ on $C_3^{(+)}$ (size 6), and $[4, 4, 4, 4, 4, 4]$ on $C_6^{(+)}$ (size 8); hence $\langle Z_i \rangle = 0$ for all k . Logical action: $\bar{U} = \text{diag}(1, \omega_9^3, \omega_9^6)$ (order $9/\gcd(9, 3, 6) = 3$).

6 Beyond nondegenerate residues

So far, we have specialized the SSLP framework to distance $d = 2$ and imposed two simplifying guards to make the search tractable: (i) residue nondegeneracy, where each logical state $|j_L\rangle$ is supported on a distinct residue class $C_{S_j}(\mathbf{w})$; and (ii) a strict classical union distance $d(C) = 2$. These constraints reduced the problem of finding codes with transversal diagonal gates to a tractable set of combinatorial and linear checks. In this regime, the X/Y off-diagonals in the KL conditions vanish by residue bookkeeping, and the remaining Z -marginal equalities collapse to a compact linear feasibility test, enabling the systematic discovery of the codes reported in Section 4.

We now demonstrate the framework’s flexibility by relaxing these constraints. Removing either guard enlarges the design space and changes how the KL equations are enforced. When several logical states share the same residue class, the transversal gate $U(\mathbf{w}, m)$ induces a degenerate logical action on that subspace. Within such a degenerate block, the KL conditions no longer vanish automatically and must be satisfied through explicit cancellations. For $j \neq k$ within the same residue block r , the conditions become:

$$\sum_{x \in C_r(\mathbf{w})} (-1)^{x_i} a_x^{(j)*} a_x^{(k)} = 0, \quad \sum_{x \in C_r(\mathbf{w})} a_x^{(j)*} a_{x \oplus e_i}^{(k)} = 0, \quad \sum_{x \in C_r(\mathbf{w})} i(-1)^{x_i} a_x^{(j)*} a_{x \oplus e_i}^{(k)} = 0,$$

for Z_i, X_i, Y_i respectively and all $j \neq k$ in the block, while the diagonals require $\sum_x (1 - 2x_i) |a_x^{(j)}|^2$ to be independent of j . If we drop the guard $d(C) = 2$, Hamming-1 neighbors can occur within the union support, so X_i and Y_i no longer vanish “for free”.

With these relaxations, it can be harder for the agent to run the search and return analytic forms; nevertheless, the agent still finds concrete examples. As an illustration of this richer design space, we tasked the agents with constructing a code implementing a non-trivial two-qubit logical gate, a controlled-phase gate, which requires degeneracy in the logical phases. We target and found a $((6, 4, 2))$ code with transversal controlled-phase gate $\text{diag}(1, 1, 1, i)$. The agent-constructed example takes $m = 4$ and $\mathbf{w} = (1, 3, 2, 2, 2, 2)$ and intentionally places $|0_L\rangle, |1_L\rangle, |2_L\rangle$ in the same residue class with residue value 0, with $|3_L\rangle$ in residue value 1. Inside the residue-0 block we choose the character signs $s_0 = 1, s_1 = \chi_3\chi_4, s_2 = \chi_3\chi_5$, indexed by $t \in \mathbb{F}_2^3$ over the even-parity subset of the last four qubits. This structure ensures all Z_i off-diagonals among $|0_L\rangle, |1_L\rangle, |2_L\rangle$ vanish by orthogonality, while residue separation and parity structure handle the X/Y terms against $|3_L\rangle$. Direct computation confirms that every weight-1 Pauli satisfies $\langle j_L | E | k_L \rangle = 0$ for $j \neq k$ and that the transversal

$$U = \bigotimes_{j=1}^6 Z\left(\frac{\pi}{2} w_j\right) \quad \text{acts as} \quad U_L = \text{diag}(1, 1, 1, i).$$

This instance lies beyond the scope of our initial systematic search, demonstrating that relaxing the simplifying guards can yield genuinely new codes and logical gates.

To be more precise, we take modulus $m = 4$ and $\mathbf{w} = (1, 3, 2, 2, 2, 2) \in \mathbb{Z}_4^6$. The residue of a bit string $x = (x_1, \dots, x_6)$ is

$$\text{res}(x) \equiv \mathbf{w} \cdot x \pmod{4} = x_1 - x_2 + 2(x_3 \oplus x_4 \oplus x_5 \oplus x_6) \pmod{4}.$$

We structure the code using the even and odd parity subsets of the last four bits, indexed by $t = (t_1, t_2, t_3) \in \mathbb{F}_2^3$ via the maps:

$$\phi(t) = (t_1, t_2, t_3, t_1 \oplus t_2 \oplus t_3) \quad (\text{even parity}), \quad \psi(t) = (t_1, t_2, t_3, 1 \oplus t_1 \oplus t_2 \oplus t_3) \quad (\text{odd parity}).$$

Write the $\{\pm 1\}$ -valued characters $\chi_3(t) = (-1)^{t_1}, \chi_4(t) = (-1)^{t_2}, \chi_5(t) = (-1)^{t_3}$. On the even-parity subset we have $(-1)^{v_3} = \chi_3, (-1)^{v_4} = \chi_4, (-1)^{v_5} = \chi_5$, and $(-1)^{v_6} = \chi_3\chi_4\chi_5$.

We define four orthonormal logical states. Three states, $|0_L\rangle, |1_L\rangle, |2_L\rangle$, are supported on strings with residue 0, while $|3_L\rangle$ is supported on strings with residue 1. All coefficients have magnitude $\frac{1}{4}$. For the degenerate block, we introduce signs based on characters:

$$s_0(t) = 1, \quad s_1(t) = \chi_3(t)\chi_4(t), \quad s_2(t) = \chi_3(t)\chi_5(t).$$

The logical states are then defined as:

$$|j_L\rangle = \frac{1}{4} \sum_{t \in \mathbb{F}_2^3} s_j(t) (|00\phi(t)\rangle + |11\phi(t)\rangle) \quad \text{for } j = 0, 1, 2,$$

$$|3_L\rangle = \frac{1}{4} \sum_{t \in \mathbb{F}_2^3} \chi_5(t) (|10\phi(t)\rangle + |01\psi(t)\rangle).$$

For completeness, we expand the four codewords explicitly in the computational basis $|x_1x_2v_3v_4v_5v_6\rangle$:

$$\begin{aligned}
|0_L\rangle &= \frac{1}{4} \left(|000000\rangle + |001001\rangle + |000101\rangle + |000011\rangle + |001100\rangle + |001010\rangle + |000110\rangle + |001111\rangle \right. \\
&\quad \left. + |110000\rangle + |111001\rangle + |110101\rangle + |110011\rangle + |111100\rangle + |111010\rangle + |110110\rangle + |111111\rangle \right), \\
|1_L\rangle &= \frac{1}{4} \left(|000000\rangle - |001001\rangle - |000101\rangle + |000011\rangle + |001100\rangle - |001010\rangle - |000110\rangle + |001111\rangle \right. \\
&\quad \left. + |110000\rangle - |111001\rangle - |110101\rangle + |110011\rangle + |111100\rangle - |111010\rangle - |110110\rangle + |111111\rangle \right), \\
|2_L\rangle &= \frac{1}{4} \left(|000000\rangle - |001001\rangle + |000101\rangle - |000011\rangle - |001100\rangle + |001010\rangle - |000110\rangle + |001111\rangle \right. \\
&\quad \left. + |110000\rangle - |111001\rangle + |110101\rangle - |110011\rangle - |111100\rangle + |111010\rangle - |110110\rangle + |111111\rangle \right), \\
|3_L\rangle &= \frac{1}{4} \left(|100000\rangle + |010001\rangle + |101001\rangle + |011000\rangle + |100101\rangle + |010100\rangle - |100011\rangle - |010010\rangle \right. \\
&\quad \left. + |101100\rangle + |011101\rangle - |101010\rangle - |011011\rangle - |100110\rangle - |010111\rangle - |101111\rangle - |011110\rangle \right).
\end{aligned}$$

Each state has norm 1 and the four are mutually orthogonal: $|3_L\rangle$ is orthogonal to the residue-0 block by disjoint support; within $\{|0_L\rangle, |1_L\rangle, |2_L\rangle\}$, orthogonality is $\frac{1}{8} \sum_t s_j(t) s_k(t) = \delta_{jk}$ by character orthogonality on $\mathbb{F}_2^3$.

We verify the distance-2 KL equations for all weight-1 Paulis $E \in \{X_i, Y_i, Z_i\}$. For $i \in \{3, 4, 5, 6\}$ a bit flip toggles the parity of the last four bits and hence shifts the residue by 2 modulo 4; therefore $X_i |m_L\rangle$ (and likewise $Y_i = iX_iZ_i$) has disjoint support from every $|j_L\rangle$, giving $\langle j_L | X_i |k_L\rangle = \langle j_L | Y_i |k_L\rangle = 0$. The Z_i diagonals vanish in each logical state by 50/50 balance of $v_i \in \{0, 1\}$; among $|0_L\rangle, |1_L\rangle, |2_L\rangle$ the off-diagonals reduce to $\frac{1}{8} \sum_t (-1)^{v_i(\phi(t))} s_j s_k = \frac{1}{8} \sum_t \chi_i s_j s_k = 0$, since with $s_0 = 1, s_1 = \chi_3 \chi_4, s_2 = \chi_3 \chi_5$ the product $\chi_i s_j s_k$ is a nontrivial character for $j \neq k$. For $i = 1$ and 2 , the operators X_i, Y_i map residue-0 supports to residue-1 supports (and vice versa), so all their matrix elements vanish by residue separation; Z_1, Z_2 have zero diagonals by the 00/11 symmetry and zero off-diagonals because contributions from the 00-even-parity half cancel those from the 11-even-parity half term by term. Thus, the KL conditions are satisfied for all single-qubit Pauli errors, and the code has distance 2.

For the transversal action, define $Z(\theta) = \text{diag}(1, e^{i\theta})$ and $U = \bigotimes_{j=1}^6 Z(\frac{\pi}{2} w_j)$. On $|x\rangle$ this contributes a phase $e^{i\frac{\pi}{2} \mathbf{w} \cdot x} = i^{\text{res}(x)}$, so U acts as a constant phase on each residue class. Since $|0_L\rangle, |1_L\rangle, |2_L\rangle$ lie in residue value 0 and $|3_L\rangle$ lies in residue value 1, we obtain $U_L = \text{diag}(1, 1, 1, i)$. This explicit example lies beyond the nondegenerate-residue screen of our strict pipeline (three logical states share residue value 0), illustrating how relaxing that guard yields genuinely new codes with nontrivial diagonal transversals while keeping the KL equations fully satisfied.

7 Discussion and Outlook

We have developed an automated discovery pipeline that systematically finds quantum error-correcting codes with prescribed transversal gates, producing a broad, certificate-backed catalog for distance-2 codes on small qubit numbers. The workflow, built on TeXRA [22] with GPT-5 [23], combines three specialized agents—Synthesis, Search, and Audit—under human orchestration to explore parameter spaces, convert numerical solutions to exact rational forms, and verify all results independently. For $K \in \{2, 3, 4\}$ logical dimensions on $n \leq 6$ qubits, we present a collection of *new* (previously unreported) codes realizing cyclic transversal gate orders from 2 to 18, each certified with exact KL equations [28] and explicit amplitudes whose square are rational, with many more to come easily. Beyond this enumeration, the Synthesis Agent extracted infinite analytical families with closed-form constructions, demonstrating how computational discovery feeds mathematical generalization.

Our multi-agent architecture have been motivated from practical challenges. Finite context windows [23, 13] can become contaminated with errors that propagate through reasoning: a mistaken factor may bias later calculations, and models often exhibit anchoring behavior, defending prior results rather than reconsidering them [53]. Asking a model to audit its own work proved less effective than using an independent agent. A separate Audit Agent operating without access to Search Agent outputs improved error detection considerably, catching calculation mistakes, sign errors, and logical inconsistencies that self-checks sometimes missed. The Audit Agent performed both computational verification—generating independent code to check numerical instances—and analytical reasoning, validating the mathematical derivations of the closed-form families presented in Sec. 5. However, audit agents can exhibit inconsistent

focus across runs, requiring multiple passes. This separation mirrors the software engineering principle of independent testing [16]. A second obstacle was notational drift across the project timeline. Different agents in separate sessions naturally developed slightly incompatible conventions: conflicting definitions, inconsistent indexing, and verbose repetition. Specialized agents with explicit prompts for notation-checking and redundancy removal resolved many conflicts automatically, though ambiguous cases required human decisions.

Human oversight proved essential throughout: maintaining clean conversation starts prevented context pollution; providing concrete examples enabled validation of abstract claims (without examples, models occasionally generated exceedingly formal but unclear generalizations); and manuscript preparation required substantial editing for coherence and notational uniformity. Critically, all new mathematical results-discovered codes, analytical families, closed-form proofs-were generated by agents. The human role was supervisory: validating correctness, guiding strategies, and ensuring presentation, but not producing discoveries. We were motivated whether such a workflow could yield rigorous new results, and our experience suggests this mode is viable for problems with appropriate structure. The text reflects this curation, preserving agent-generated content while imposing coherence unsuitable for raw outputs.

The technical key decision is recognizing together with the agent that the SSLP framework [49] reduces distance-2 feasibility to tractable subproblems (to impose the *classical union distance* condition $d(C) = 2$). The technical key decision—made together with the agents—was to impose the *classical union distance* condition $d(C) = 2$ within the SSLP setup for the distance-2 case. Modular residue classes partition computational basis strings so that transversal diagonal gates induce predictable logical phases. When each logical state occupies a distinct residue class, single-qubit X and Y errors—which flip bits and shift residues—automatically satisfy off-diagonal KL constraints by disjoint support. The remaining Z -marginal conditions become linear equations on probability amplitudes, solvable via standard LP methods. This reformulation transforms a nonlinear, high-dimensional feasibility problem into discrete screening (residue compatibility) plus convex optimization (probability matching)—a structure enabling systematic enumeration and exact analytical reconstruction through continued-fraction approximation [61, 60] and integer-preserving projection [58].

While our application-distance-2 codes with diagonal transversals-is specific, the underlying methodology establishes a general paradigm for AI-assisted discovery in theoretical physics and mathematics. The essential ingredients are: (1) mathematical reformulation exposing tractable substructure (here, residue separation plus linear constraints); (2) multi-agent orchestration [19, 20] with specialized roles (problem formulation, systematic search, independent verification); (3) tight human-AI feedback loops where humans provide domain insight and validation while agents execute scale exploration; and (4) problems possessing verifiable structure where solutions are hard to generate but easy to check [4, 5]. These conditions appear widely in classification problems across mathematical physics-characterizing symmetry-protected phases [62, 63], enumerating lattice models with dualities, finding exactly solvable models in condensed matter physics, discovering integrable structures-domains where systematic exploration meets pattern recognition.

The workflow’s architecture demonstrates productive human-AI collaboration. The Synthesis Agent operates through derivation-then-edit workflows [51, 52] that expand mathematical reasoning in internal scratchpads before producing formal outputs. From a single worked example, it deduced the combinatorial reformulation and proposed screening algorithms. The Search Agent uses tool-use loops [24, 12] to iteratively generate code, execute searches, and process results, handling computational tasks at scales infeasible manually. The Audit Agent operates behind a deliberate no-communication barrier, independently verifying every instance to prevent error propagation—a protocol ensuring mathematical rigor despite AI fallibility. This separation of concerns-formulation, execution, verification-mirrors successful software engineering practices [16] adapted for mathematical discovery.

For quantum error correction specifically, our catalog enlarges the known nonadditive design space [34, 46] beyond prior constructions [35, 38, 47]. The long-term objective is a principled classification of transversal groups attainable by small codes, moving beyond stabilizer-only catalogs [45]—which leave substantial nonstabilizer (nonadditive) structure unaccounted for—to a general treatment of small codes. Codes with high-order transversal gates may enable efficient fault-tolerant protocols via magic-state distillation [64] and gadget-based universality [32]. The $((6, 4, 2))$ controlled-phase code, constructed by relaxing the distinct-residue (“nondegenerate-residue”) assumption, demonstrates richer design possibilities where degenerate logical actions combine with character-theoretic cancellations. Extending this approach to non-Abelian transversal groups remains an open direction; SSLP-like reformulations could plausibly yield new constructions.

Methodologically, this work opens a paradigm in which AI systems augment theoretical science through

structured exploration and analytical pattern extraction [1, 9]. The success derived from matching AI capabilities-tool use, symbolic manipulation [11, 17], reasoning workflows [18]-to problem structure (discrete search, verifiable constraints, extractable patterns). Identifying other domains with similar properties and refining orchestration protocols will extend this approach across mathematical physics. Our workflow demonstrates that systematic discovery, previously requiring expert intuition developed over long period of time can become an automated analytical pipeline when mathematical structure meets multi-agent AI orchestration.

Acknowledgements

The authors would like to express their gratitude to Alexander Frei and Ningping Cao for helpful discussions.

References

- [1] H. Wang, T. Fu, Y. Du, W. Gao, K. Huang, Z. Liu, P. Chandak, S. Liu, P. Van Katwyk, A. Deac, A. Anandkumar, K. Bergen, C. P. Gomes, S. Ho, P. Kohli, J. Lasenby, J. Leskovec, T.-Y. Liu, A. Manrai, D. Marks, B. Ramsundar, L. Song, J. Sun, J. Tang, P. Veličković, M. Welling, L. Zhang, C. W. Coley, Y. Bengio, and M. Zitnik, “Scientific discovery in the age of artificial intelligence,” *Nature* **620**, 47 (2023).
- [2] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha, “The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery,” (2024), [arXiv:2408.06292 \[cs.AI\]](#) .
- [3] M. Binz, S. Alaniz, A. Roskies, B. Aczel, C. T. Bergstrom, C. Allen, D. Schad, D. Wulff, J. D. West, Q. Zhang, R. M. Shiffrin, S. J. Gershman, V. Popov, E. M. Bender, M. Marelli, M. M. Botvinick, Z. Akata, and E. Schulz, “How should the advancement of large language models affect the practice of science?” *Proc. Natl. Acad. Sci.* **122**, e2401227121 (2025).
- [4] E. Glazer, E. Erdil, T. Besiroglu, D. Chicharro, E. Chen, A. Gunning, C. F. Olsson, J.-S. Denain, A. Ho, E. d. O. Santos, O. Järvinen, M. Barnett, R. Sandler, M. Vrzala, J. Sevilla, Q. Ren, E. Pratt, L. Levine, G. Barkley, N. Stewart, B. Grechuk, T. Grechuk, and S. V. Enugandla, “FrontierMath: A benchmark for evaluating advanced mathematical reasoning in AI,” (2024), [arXiv:2411.04872 \[cs.AI\]](#) .
- [5] S. Lu, Z. Jin, T. J. Zhang, P. Kos, J. I. Cirac, and B. Schölkopf, “Can Theoretical Physics Research Benefit from Language Agents?” (2025), [arXiv:2506.06214 \[cs.CL\]](#) .
- [6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of Go with deep neural networks and tree search,” *Nature* **529**, 484 (2016).
- [7] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, “Mastering the game of Go without human knowledge,” *Nature* **550**, 354 (2017).
- [8] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, “Highly accurate protein structure prediction with AlphaFold,” *nature* **596**, 583 (2021).
- [9] B. Romera-Paredes, M. Barekatin, A. Novikov, M. Balog, M. P. Kumar, E. Dupont, F. J. R. Ruiz, J. S. Ellenberg, P. Wang, O. Fawzi, P. Kohli, and A. Fawzi, “Mathematical discoveries from program search with large language models,” *Nature* **625**, 468 (2024).
- [10] T. H. Trinh, Y. Wu, Q. V. Le, H. He, and T. Luong, “Solving olympiad geometry without human demonstrations,” *Nature* **625**, 476 (2024).

- [11] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language Models are Few-Shot Learners,” in *Adv. Neural Inf. Process. Syst.*, Vol. 33 (Curran Associates, Inc., 2020) pp. 1877–1901, [arXiv:2005.14165 \[cs.CL\]](#) .
- [12] T. Schick, J. Dwivedi-Yu, R. Dessi, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, “Toolformer: Language Models Can Teach Themselves to Use Tools,” in *Thirty-Seventh Conference on Neural Information Processing Systems* (2023).
- [13] Anthropic, “[Introducing the model context protocol](#),” (2024).
- [14] S. G. Patil, T. Zhang, X. Wang, and J. E. Gonzalez, “Gorilla: Large language model connected with massive apis,” in *Advances in Neural Information Processing Systems*, Vol. 37 (2024) pp. 126544–126565.
- [15] M. Tian, L. Gao, D. Zhang, X. Chen, C. Fan, X. Guo, R. Haas, P. Ji, K. Krongchon, Y. Li, S. Liu, D. Luo, Y. Ma, H. Tong, K. Trinh, C. Tian, Z. Wang, B. Wu, S. Yin, M. Zhu, K. Lieret, Y. Lu, G. Liu, Y. Du, T. Tao, O. Press, J. Callan, E. A. Huerta, and H. Peng, “SciCode: A Research Coding Benchmark Curated by Scientists,” in *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track* (2024).
- [16] C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, and K. R. Narasimhan, “SWE-bench: Can language models resolve real-world github issues?” in *The Twelfth International Conference on Learning Representations* (2024).
- [17] OpenAI, “[OpenAI O1](#),” (2024).
- [18] D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Xu, H. Ding, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Chen, J. Yuan, J. Tu, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. You, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Zhou, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. Q. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, and Z. Zhang, “DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning,” *Nature* **645**, 633 (2025).
- [19] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, and C. Wang, “[AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation](#),” (2023), [arXiv:2308.08155 \[cs.AI\]](#) .
- [20] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, “[Improving Factuality and Reasoning in Language Models through Multiagent Debate](#),” (2023), [arXiv:2305.14325 \[cs.CL\]](#) .
- [21] T. Sumers, S. Yao, K. Narasimhan, and T. Griffiths, “Cognitive Architectures for Language Agents,” *Transactions on Machine Learning Research* (2023).
- [22] TeXRA, “[TeXRA: Your intelligent academic research assistant](#),” (2025).
- [23] OpenAI, “[GPT-5 model \(API documentation\)](#),” (2025).

- [24] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, “ReAct: Synergizing Reasoning and Acting in Language Models,” in *The Eleventh International Conference on Learning Representations* (2023).
- [25] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2010).
- [26] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, “Quantum error correction via codes over GF (4),” *IEEE Trans. Inf. Theory* **44**, 1369 (1998).
- [27] A. M. Steane, “Error correcting codes in quantum theory,” *Physical Review Letters* **77**, 793 (1996).
- [28] E. Knill and R. Laflamme, “Theory of quantum error-correcting codes,” *Phys. Rev. A* **55**, 900 (1997).
- [29] B. Zeng, A. W. Cross, and I. L. Chuang, “Transversality versus universality for additive quantum codes,” *IEEE Trans. Inf. Theory* **57**, 6272 (2011).
- [30] B. Eastin and E. Knill, “Restrictions on Transversal Encoded Quantum Gate Sets,” *Phys. Rev. Lett.* **102**, 110502 (2009).
- [31] Z.-W. Liu and S. Zhou, “Approximate symmetries and quantum error correction,” *npj Quantum Information* **9**, 119 (2023).
- [32] J. T. Anderson and T. Jochym-O’Connor, “Classification of transversal gates in qubit stabilizer codes,” *Quantum Information & Computation* **16**, 771 (2016).
- [33] D. Gottesman, *Stabilizer Codes and Quantum Error Correction*, Ph.D. thesis, California Institute of Technology (1997).
- [34] E. M. Rains, R. H. Hardin, P. W. Shor, and N. J. A. Sloane, “A Nonadditive Quantum Code,” *Phys. Rev. Lett.* **79**, 953 (1997).
- [35] A. W. Cross, G. Smith, J. Smolin, and B. Zeng, “Codeword stabilized quantum codes,” *IEEE Trans. Inf. Theory* **55**, 433 (2009).
- [36] I. L. Chuang, A. W. Cross, G. Smith, J. Smolin, and B. Zeng, “Codeword stabilized quantum codes: Algorithm and structure,” *J. Math. Phys.* **50**, 42109 (2009).
- [37] S. Yu, Q. Chen, and C. H. Oh, “Graphical Quantum Error-Correcting Codes,” (2007), [arXiv:0709.1780 \[quant-ph\]](#).
- [38] S. Yu, Q. Chen, C. Lai, and C. Oh, “Nonadditive quantum error-correcting code,” *Physical review letters* **101**, 090501 (2008).
- [39] M. Grassl and M. Rotteler, “Non-additive quantum codes from goethals and preparata codes,” in *2008 IEEE Information Theory Workshop* (IEEE, 2008) pp. 396–400.
- [40] H. Pollatsek and M. B. Ruskai, “Permutationally invariant codes for quantum error correction,” *Linear Algebra and its Applications* **392**, 255 (2004).
- [41] Y. Ouyang, “Permutation-invariant quantum codes,” *Physical Review A* **90**, 062317 (2014).
- [42] Y. Ouyang, “Permutation-invariant quantum coding for quantum deletion channels,” in *2021 IEEE International Symposium on Information Theory (ISIT)* (IEEE, 2021) pp. 1499–1503.
- [43] Y. Ouyang, Y. Jing, and G. K. Brennen, “Measurement-free code-switching for low overhead quantum computation using permutation invariant codes,” *arXiv preprint arXiv:2411.13142* (2024), [arXiv:2411.13142](#).
- [44] M. Du, C. Zhang, Y.-T. Poon, and B. Zeng, “Characterizing quantum codes via the coefficients in knill-laflamme conditions,” *arXiv preprint arXiv:2410.07983* (2024), [arXiv:2410.07983](#).
- [45] A. Cross and D. Vandeth, “Small binary stabilizer subsystem codes,” *arXiv preprint arXiv:2501.17447* (2025), [arXiv:2501.17447](#).

- [46] E. M. Rains, “Quantum codes of minimum distance two,” *IEEE Transactions on Information theory* **45**, 266 (2002).
- [47] E. Kubischta and I. Teixeira, “Family of quantum codes with exotic transversal gates,” *Physical Review Letters* **131**, 240601 (2023).
- [48] E. Kubischta and I. Teixeira, “Permutation-invariant quantum codes with transversal generalized phase gates,” *IEEE Transactions on Information Theory* **71**, 485 (2024).
- [49] C. Zhang, Z. Wu, S. Huang, and B. Zeng, “Transversal gates in nonadditive quantum codes,” *arXiv preprint arXiv:2504.20847* (2025), [arXiv:2504.20847](#).
- [50] Microsoft Corporation, “[Visual studio code](#),” (2024).
- [51] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)* (2022) pp. 24824–24837.
- [52] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, “[Reflexion: Language Agents with Verbal Reinforcement Learning](#),” (2023), [arXiv:2303.11366 \[cs.AI\]](#).
- [53] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe, “[Let’s Verify Step by Step](#),” (2023), [arXiv:2305.20050 \[cs.LG\]](#).
- [54] F. Tilmann and C. Topping, “[Latexdiff: A Perl script for visual mark up and revision of significant differences between two LaTeX files](#),” (2024).
- [55] C. Carathéodory, “Über den Variabilitätsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen,” *Mathematische Annalen* **64**, 95 (1907).
- [56] E. H. Bareiss, “Sylvester’s identity and multistep integer-preserving gaussian elimination,” *Mathematics of Computation* **22**, 565 (1968).
- [57] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization* (Athena Scientific, 1997).
- [58] A. Schrijver, *Theory of Linear and Integer Programming* (Wiley, 1986).
- [59] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra* (Cambridge University Press, 2003).
- [60] A. K. Lenstra, H. W. Lenstra, and L. Lovász, “Factoring polynomials with rational coefficients,” *Mathematische Annalen* **261**, 515 (1982).
- [61] H. R. P. Ferguson, D. H. Bailey, and S. Arno, “Analysis of PSLQ, an integer relation finding algorithm,” *Mathematics of Computation* **68**, 351 (1999).
- [62] X. Chen, Z.-C. Gu, Z.-X. Liu, and X.-G. Wen, “Symmetry protected topological orders and the group cohomology of their symmetry group,” *Phys. Rev. B* **87**, 155114 (2013).
- [63] B. Zeng, X. Chen, D. Zhou, and X.-G. Wen, *Quantum Information Meets Quantum Matter – From Quantum Entanglement to Topological Phase in Many-Body Systems*, 1st ed., Quantum Science and Technology (Springer-Verlag New York, 2019) [arXiv:1508.02595 \[quant-ph\]](#).
- [64] S. Bravyi and A. Kitaev, “Universal quantum computation with ideal Clifford gates and noisy ancillas,” *Phys. Rev. A* **71**, 022316 (2005).