

Program-Synthesis-Driven Autodesign of Universal Unitary Operators

Yifei Zhang^{1,2,*} Dong Chen^{3,*} Fan Wang^{4,2} Wenrui Zhang⁵ Yan Chen,⁵ Dingding Han,^{6,7} Jianmin Yuan⁸,
Xiangjin Kong^{1,2,†} and Yu-Gang Ma^{1,2,9,‡}

¹Key Laboratory of Nuclear Physics and Ion-beam Application (MOE), Institute of Modern Physics,
Fudan University, Shanghai 200433, China

²Research Center for Theoretical Nuclear Physics, NSFC and Fudan University, Shanghai 200438, China

³Huawei Technologies Co., Ltd, Beijing 100095, China

⁴Department of Industrial Engineering and Decision Analytics, Hong Kong University of Science and
Technology, HongKong, China

⁵Hunan Key Laboratory of Mechanism and Technology of Quantum Information, Changsha 410073, China

⁶School of Information Science and Technology, Fudan University, Shanghai 200433, China

⁷Research Institute of Intelligent Complex Systems, Fudan University, Shanghai 200433, China

⁸Institute of Atomic and Molecular Physics, Jilin University, Changchun 130012, China

⁹School of Physics, East China Normal University, Shanghai 200062, China

 (Received 8 January 2026; revised 15 April 2026; accepted 9 June 2026; published 20 July 2026)

We demonstrate that AI-driven program synthesis can autonomously discover fundamental strategies for decomposing unitary matrices in photonic networks. By extending DreamCoder to complex-valued linear algebra, the system generates decomposition programs achieving the minimal $N(N-1)/2$ Mach-Zehnder interferometers, distinct from both Reck and Clements architectures. Learned programs encode dimension-agnostic invariants: strategies discovered for 5×5 matrices generalize to higher dimensions such as 64×64 . The discovered programs encode interpretable, dimension-agnostic construction rules. These rules generalize across matrix sizes without retraining, demonstrating that autonomous program synthesis can serve as a scalable paradigm for algorithm discovery and the automated design of universal unitary operators. Beyond universal decompositions, the system automatically exploits matrix structure to reduce the interferometer count below the universal theoretical bound. For instance, for Householder matrices, it discovers a dimension-independent rule that requires only $2N-3$ MZIs. This achieves linear, rather than quadratic, scaling and generalizes to arbitrary N without retraining. For matrices obtained from the singular value decomposition of sparse matrices, reductions generally increase with sparsity, reaching up to 38% fewer MZIs than the universal theoretical bound $N(N-1)/2$ at 95% sparsity. These MZI reductions translate directly into practical hardware benefits for scalable photonic implementations. Taken together, the system functions as a single unified engine that discovers both universal decomposition rules and matrix-specific optimizations, without being provided with the structural or analytical properties of the input matrices.

DOI: [10.1103/49c3-4rp4](https://doi.org/10.1103/49c3-4rp4)

Unitary transformations lie at the foundation of quantum mechanics, governing the evolution of quantum states in closed systems [1], and are central to modern photonic computing and quantum information processing [2–4]. In linear optical systems, unitary matrices determine how multimode fields interfere and propagate through reconfigurable interferometric networks, allowing direct hardware implementations of matrix operations with ultrahigh speed, low energy consumption, and intrinsic parallelism [4,5]. These capabilities make photonic platforms attractive for optical neural networks [6–11], where unitaries implement trainable linear layers, and for quantum computation

[2,12–15], where they realize universal linear-optical circuits. Rapid advances in integrated photonics have produced large-scale, low-loss, and programmable interferometer meshes [16–18], motivating scalable and systematic methods to realize arbitrary high-dimensional unitaries in hardware.

In practice, arbitrary $N \times N$ unitary transformations are implemented by decomposing them into sequences of Mach-Zehnder interferometers (MZIs), which serve as universal two-dimensional beam-splitter gates [19,20]. Mathematically, any matrix can be expressed via singular value decomposition (SVD) as $W = U\Sigma V^\dagger$ [21], producing two unitary matrices U and $V^\dagger$ and a diagonal matrix Σ . Since diagonal matrices Σ can be directly realized with MZIs, the challenge reduces to efficiently decomposing the unitary components U and $V^\dagger$. Classical schemes, such as the Reck triangular mesh [22] and the Clements rectangular mesh [23], provide hand-derived constructions

*These authors contributed equally to this work.

†Contact author: kongxiangjin@fudan.edu.cn

‡Contact author: mayugang@fudan.edu.cn

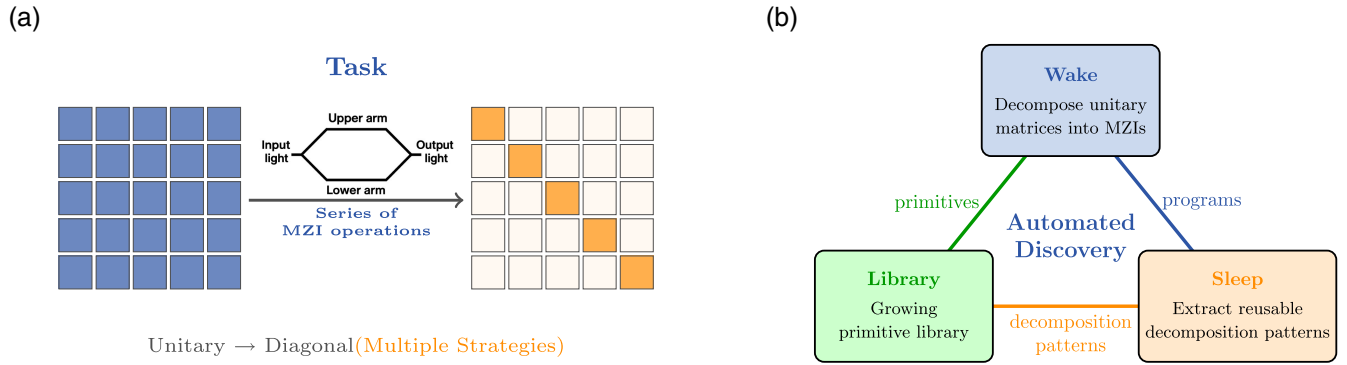


FIG. 1. Autonomous discovery of unitary decomposition strategies. (a) Task transformation: unitary matrices are decomposed into diagonal form through a series of MZI operations, with multiple decomposition strategies available. (b) DreamCoder’s abstraction loop: candidate MZI programs are enumerated and tested on decomposition tasks, recurring fragments from successful programs are compressed into reusable library primitives, and the enlarged library is reused to bias subsequent searches toward deeper decompositions.

using $N(N-1)/2$ MZIs, forming the algorithmic backbone for universal linear optics and underpinning most present-day photonic processors [7,12].

Despite their foundational role, these schemes reflect specific human-designed elimination strategies derived from analytic reasoning and occupy only a small region of the combinatorial space of possible elimination orderings, leaving open the question of whether alternative, equally fundamental strategies exist. To address this, we employ automated reasoning to systematically explore this landscape, uncovering strategies that extend beyond human intuition and yield generalizable constructions. In pursuit of this goal, we develop a synthesis engine—built on the DreamCoder framework [24] and extended to manipulate complex-valued linear algebra—that autonomously discovers strategies for decomposing unitary matrices. The synthesized programs are not merely confined to known triangular or rectangular meshes. Instead, the system identifies previously unreported elimination orderings that achieve the optimal $N(N-1)/2$ MZIs. Analysis of the learned programs reveals that these constructions encode dimension-agnostic algorithmic regularities: strategies discovered on 5×5 matrices generalize to arbitrary higher dimensions such as 64×64 without retraining, indicating that the system has internalized domain-general patterns rather than dimension-specific heuristics. These results establish program synthesis as a means of autonomously identifying algorithmic structure beyond known human-designed schemes, enabling the discovery of interpretable and generalizable rules for universal unitary decompositions. Beyond universal constructions, the same synthesis engine adapts to structured inputs, automatically discovering structure-specific reductions independent of any prior assumptions about the matrix structure. For analytically structured matrices such as Householder reflectors, the system discovers a dimension-independent rule requiring only $2N-3$ MZIs, reducing complexity from quadratic to

linear scaling, and generalizes to arbitrary N without retraining. For matrices obtained from the SVD of sparse matrices, it uncovers structure-dependent reductions that tend to increase with sparsity, reaching up to 38% fewer MZIs than the universal theoretical bound $N(N-1)/2$ at 95% sparsity. Together, these results position program synthesis as a unified framework for both discovering generalizable decomposition rules and exploiting matrix-specific structure, enabling scalable automated design of photonic unitary circuits.

As shown in Fig. 1(a), our learning objective is to discover programs that transform arbitrary unitary matrices into diagonal form. This formulation directly corresponds to photonic implementation: synthesizing an arbitrary unitary matrix U on a photonic chip decomposes into two steps—an MZI cascade network P and a diagonal matrix D such that $U = P^\dagger D$. Since D requires no algorithmic search, our learning objective simplifies to finding the sequence P that performs the diagonalization. Training tasks consist of randomly generated unitary matrices U_k [25]. A program p transforms U_k into a diagonal matrix $D_k = p(U_k)$, where D_k retains phase information $e^{i\phi_j}$ on its diagonals. Success is determined solely by the diagonalization constraint: specifically, all off-diagonal elements ($i \neq j$) must satisfy $|p(U_k)_{ij}| < \epsilon_{\text{elem}}$, where $\epsilon_{\text{elem}} = 5 \times 10^{-4}$ in the simulation. The specific value of this threshold does not affect program acceptance, as successfully synthesized programs typically drive residuals down to machine precision ($\sim 10^{-16}$). Meanwhile, the diagonal phase values in D_k are not prescribed, but emerge naturally from the decomposition process.

To solve this diagonalization task, we define a set of $N(N-1)$ MZI primitives $\{R_{ij}, L_{ij}\}$ for $i > j$, with subscripts denoting the target element position at row i , column j . Each primitive eliminates one matrix element. The decomposition strategy exploits the unitarity constraint $U^\dagger U = I$, where systematic elimination of lower-triangular

elements automatically enforces upper-triangular nullity, yielding a diagonal matrix. Each MZI primitive corresponds to a beam splitter transformation acting on adjacent modes, eliminating matrix element u_{ij} via adaptive phase shifts. For R_{ij} -type primitives, the transformation eliminates element u_{ij} by mixing columns j and $j + 1$ via right multiplication. From the nullification condition $(U R_{ij})_{ij} = 0$, the required phase θ_{ij} and mixing angle ω_{ij} are analytically derived: the ratio $r = u_{ij}/u_{i,j+1}$ is computed, with $\theta_{ij} = \arg(r)$ compensating the phase difference and $\omega_{ij} = \arctan(|r|)$ determining the mixing ratio. This constructs an $N \times N$ transformation matrix by embedding the 2×2 MZI block

$$M_{R,ij} = \begin{pmatrix} e^{-i\theta_{ij}} \cos \omega_{ij} & e^{-i\theta_{ij}} \sin \omega_{ij} \\ -\sin \omega_{ij} & \cos \omega_{ij} \end{pmatrix} \quad (1)$$

at positions $(j, j + 1)$ within an identity matrix, applying it via right multiplication $U' = U R_{ij}$. L_{ij} -type primitives similarly mix rows i and $i - 1$ via left multiplication, deriving parameters from condition $(L_{ij} U)_{ij} = 0$ and embedding the 2×2 MZI block

$$M_{L,ij} = \begin{pmatrix} e^{i\theta_{ij}} \cos \omega_{ij} & -\sin \omega_{ij} \\ e^{i\theta_{ij}} \sin \omega_{ij} & \cos \omega_{ij} \end{pmatrix} \quad (2)$$

at positions $(i - 1, i)$ within an identity matrix, applying it via left multiplication $U' = L_{ij} U$. The complete $N \times N$ matrix forms of R_{ij} and L_{ij} primitives are provided in Supplemental Material (SM) [26]. Each primitive takes a complex matrix as input and returns a transformed matrix, so primitives can be freely chained into sequences: e.g., a 3×3 decomposition might be $(\text{lambda } (R21 (L10 (L20 \$0))))$ [applied in sequence to positions $(2,0)$, $(1,0)$, and $(2,1)$], where $\$0$ denotes the input matrix and operations are applied from innermost to outermost.

Given the task definition and primitive operations, we employ program synthesis to autonomously discover effective decomposition strategies. We adopt the DreamCoder framework [24], which operates through an iterative wake-sleep cycle, as illustrated in Fig. 1(b). During the wake phase, the system searches over candidate MZI operation sequences using the current primitive library and evaluates them against decomposition tasks. During the sleep phase, it identifies recurring patterns across successful solutions and abstracts them into new reusable primitives, expanding the library. The updated library is then carried into the next wake phase, where the richer primitive vocabulary enables more efficient search over deeper program structures [27]. We extend DreamCoder to the complex-valued matrix decomposition domain; implementation details are provided in SM [26]. The MZI primitives $\{R_{ij}, L_{ij}\}$ encode physically meaningful beam-splitter transformations with analytically derived phase and rotation angles. This design can be generalized to a broad class of quantum information processing tasks

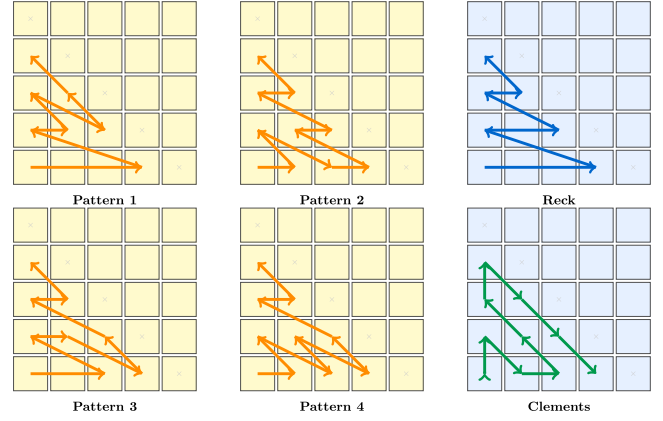


FIG. 2. Decomposition patterns discovered for 5×5 unitary matrices compared with classical reference schemes. Left four panels (orange paths, yellow background) show representative discovered patterns exhibiting diverse elimination orderings. Right two panels show classical reference schemes for comparison: Reck (top, blue path) uses bottom-up row-wise elimination, while Clements (bottom, green path) employs symmetric alternating diagonal pattern. All patterns achieve the theoretical minimum of 10 MZI operations. The discovered patterns demonstrate that the system learned flexible decomposition strategies different from classical approaches.

involving matrix operations, such as quantum circuit synthesis [28,29].

Verification of the synthesized programs depends on their target application. Universal decompositions are evaluated on multiple randomly generated unitary matrices, whereas structure-specific programs (e.g., for sparse matrices) are tested directly on their original inputs. In all cases, successfully verified programs achieve off-diagonal residuals at machine precision ($\sim 10^{-16}$). Furthermore, we validate the cross-dimensional generalization of universal programs by applying the learned patterns to larger matrices, confirming their ability to scale to arbitrary N .

Having established the framework, we now demonstrate its effectiveness on unitary decomposition tasks. To assess the system's capacity for autonomous algorithmic discovery, we employ independent learning where the system constructs complete decompositions from scratch without prior knowledge of classical schemes. This approach ensures discovered strategies emerge from the search process itself rather than reflecting biases from known methods. We first investigate the 5×5 case using independent learning within tractable search spaces. After nine wake-sleep iterations, the system autonomously discovered multiple distinct decomposition strategies, all achieving the theoretical minimum of 10 MZI operations. Figure 2 visualizes four random decomposition patterns, where the orange paths trace the sequence of operations through the matrix. Each pattern shows a different execution ordering that systematically eliminates lower-triangular elements, confirming the system learned flexible decomposition strategies rather than a single fixed

pattern. These strategies exhibit novel elimination orderings distinct from both Reck and Clements schemes, as presented in Fig. 2. The learned decomposition strategies are not tied to specific unitary instances but apply universally to arbitrary unitary matrices, functioning as generalizable algorithmic rules. To the best of our knowledge, such strategies have not been reported previously. Additional representative examples are provided in Table S1 in SM [26].

Among the discovered programs, the one ranked highest by the system, which favors solutions that are both correct and structurally simple (i.e., expressible more compactly in terms of the learned library operations) [30,31] is $(\text{lambda } (R10 (R21 (R20 (R32 (R31 (R30 (R43 (R42 (R41 (L40 \$0))))))))))$. This program corresponds to the matrix equation:

$$D = L_{40}UR_{41}R_{42}R_{43}R_{30}R_{31}R_{32}R_{20}R_{21}R_{10}, \quad (3)$$

where D is the resulting diagonal matrix. Inverting this relation yields the unitary decomposition

$$U = L_{40}^{-1}DR_{10}^{-1}R_{21}^{-1}R_{20}^{-1}R_{32}^{-1}R_{31}^{-1}R_{30}^{-1}R_{43}^{-1}R_{42}^{-1}R_{41}^{-1}. \quad (4)$$

This program directly maps to a photonic circuit implementation, where the five optical waveguides correspond to matrix modes and each MZI block executes one primitive operation. The circuit processes light through a cascade of 10 MZI stages, achieving the theoretical minimum count. This demonstrates that the system has learned a flexible decomposition strategy, revealing that the systematic elimination of lower-triangular elements admits multiple valid operation sequences.

Critically, the learned patterns exhibit cross-dimensional reasoning. Analysis of discovered programs reveals dimension-agnostic principles: systematic elimination strategies that remain valid across matrix scales. For instance, algorithmic abstractions derived from 5×5 matrices exhibit compositional generalization: they can be directly applied to higher-dimensional unitary matrices without retraining or modification. As an illustration, consider Pattern 4 shown in Fig. 2, which implements a row-pair interleaving strategy. Starting from the largest row index $i = N - 1$ and decrementing by two each step, the algorithm processes rows in pairs: it first emits primitive $R_{i,0}$, then interleaves primitives from two adjacent rows as $R_{i,j} \rightarrow R_{i-1,j-1}$ for $j = 1, \dots, i - 1$. When a single row remains ($i = 1$), $R_{1,0}$ closes the sequence. This fixed rule, discovered on 5×5 matrices, applies unchanged at any N and enumerates all $N(N - 1)/2$ operations exactly. Full pseudocode and generalization results are given in SM [26]. We validated this cross-dimensional generalization by testing programs discovered on 5×5 training instances against progressively larger matrices, including 8×8 , 16×16 , 32×32 , and 64×64 dimensions. All test cases achieve successful diagonalization at machine precision

($\sim 10^{-16}$), confirming that the learned decomposition strategy scales correctly to arbitrary dimensions.

While our learned patterns exhibit dimension-agnostic principles that allow 5×5 strategies to generalize, direct training on 6×6 matrices remains essential to uncover richer decomposition orderings inaccessible from smaller examples. However, this objective confronts a fundamental bottleneck: the search space grows exponentially with matrix dimension, making the direct enumeration of deep 15-MZI sequences intractable. As matrix dimension grows, both the primitive vocabulary and program depth increase, causing the search space to explode combinatorially. To address this challenge, we employ curriculum learning [32,33] by decomposing the problem into subtasks $\{T_3, T_2, T_1, T_0\}$. This progression allows the model to first master short subsequences in early stages (T_3), rapidly abstracting reusable patterns into library primitives via the sleep phase. Consequently, these consolidated abstractions effectively reduce the complexity for later stages (T_0), enabling the synthesis of deeply nested programs. Specifically, our results demonstrate that after just seven wake-sleep iterations, the system not only converged to the theoretical minimum depth but also expanded its library from 30 to 65 hierarchical abstractions. The system successfully rediscovered the Reck decomposition and identified novel patterns distinct from 5×5 independent learning. One such pattern can be expressed by the diagonalization relation $D = L_{5,1}L_{4,0}L_{5,0}UR_{5,2}R_{5,3}R_{5,4}R_{4,1}R_{4,2}R_{3,0}R_{4,3}R_{3,1}R_{2,0}R_{3,2}R_{2,1}R_{1,0}$. This demonstrates that curriculum learning effectively navigates the expanded search space to uncover both classical and genuinely new decomposition strategies.

Beyond universal decompositions, the system also discovers structure-aware reductions for specific matrix classes, without requiring any specialized analytical rules. We identify two broad categories of such structure-aware reductions.

The first category comprises analytically structured matrices, with Householder matrices serving as a prime example. Defined as $H = I - 2vv^\dagger$ (I is the identity matrix, $v \in \mathbb{C}^N$, $\|v\| = 1$), Householder matrices are unitary reflectors widely used in mathematics and physics for stable unitary transformations, including tridiagonalization of Hamiltonians and eigenvalue calculations in quantum and many-body systems [34–36]. By exploiting these inherent structural constraints, our system autonomously identifies a dimension-independent decomposition rule that strictly bounds the total MZI count to $2N - 3$. The recursive logic is as follows: extending to $N + 1$ dimensions requires prepending two operations, $L_{N,0}$ followed by $R_{N,N-1}$, to the N -dimensional sequence. At $N = 6$, the corresponding diagonalization relation is $D = L_{4,0}L_{5,0}HR_{5,4}R_{4,3}R_{3,0}R_{3,1}R_{3,2}R_{2,1}R_{1,0}$ (9 operations $= 2 \times 6 - 3$). This strategy reduces the hardware scaling from the universal quadratic bound of $N(N - 1)/2$ to a linear complexity. When evaluated at $N = 64$, this

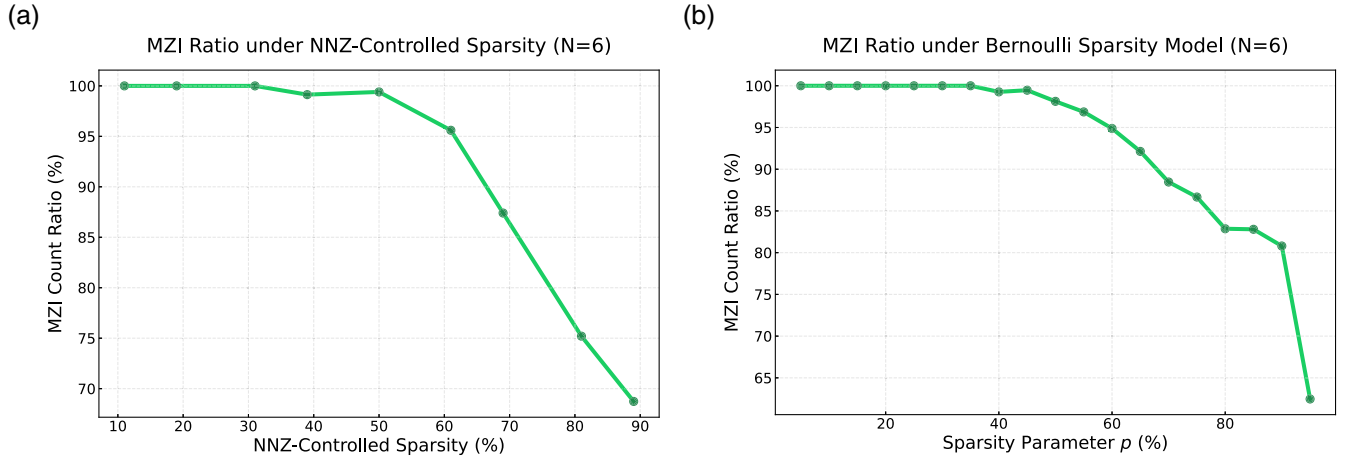


FIG. 3. MZI count ratio [normalized to the theoretical maximum $N(N-1)/2$] as a function of sparsity level for $N = 6$ matrices under NNZ-controlled and Bernoulli sparsity models. Both models show a consistent overall decrease, confirming that the MZI reduction reflects an intrinsic property of sparse unitary matrices. Each data point is averaged over nearly 100 successfully decomposed matrices per sparsity level.

generalized rule perfectly reconstructs the unitary using only 125 MZIs instead of 2016. This represents a substantial 93.8% reduction in hardware components, verified at machine precision without any retraining.

The second category involves sparse matrices. In our framework, an arbitrary input W is decomposed via SVD ($W = U\Sigma V^\dagger$), where the unitary factors U and $V^\dagger$ inherit the sparsity of W and are further decomposed into MZI sequences. Our system exploits this inherited structure to discover highly efficient MZI sequences. We validate this on $N = 6$ matrices using two sparsity models: an NNZ-controlled setting (NNZ: number of nonzero elements) with a fixed number of nonzero elements in W , and a Bernoulli model where each element of W is independently zeroed with probability p [37,38]. As shown in Fig. 3, the system achieves significant reductions relative to the universal theoretical bound $N(N-1)/2$, and these reductions generally increase with matrix sparsity. Under the NNZ-controlled model, reductions begin around 39% sparsity and reach up to 31% at 89% sparsity. Under the Bernoulli model, reductions emerge around 40% sparsity and grow more steeply at high sparsity, reaching up to 38% at 95% sparsity. This overall trend is consistent across both models, confirming that the efficiency gains reflect intrinsic structural properties of sparse unitaries rather than artifacts of specific matrix generation. For arbitrary unstructured sparse matrices, the discovered patterns are instance-specific, as there is no well-defined correspondence between sparsity patterns at different dimensions.

These MZI reductions can provide practical hardware benefits: fewer components can help reduce optical insertion loss [23], calibration complexity [39], and power consumption [40]. Although synthesized topologies may be irregular, modern photonic design flows increasingly support automated placement and routing under physical constraints such as bend radius and waveguide crossings [41,42].

In conclusion, our results show that program synthesis can uncover generalizable algorithmic structures for universal unitaries—going beyond known schemes to reveal decomposition principles and resource optimizations that classical Reck and Clements designs cannot capture. Built on a complex-valued matrix decomposition domain, our framework combines symbolic reasoning with physically grounded MZI primitives to systematically generate and verify decomposition programs, providing a scalable foundation that can be extended to broader quantum information tasks. These automatically generated constructions encode systematic construction methods, demonstrating the potential of AI-driven algorithm discovery to reveal fundamental principles and offering a powerful new approach for advancing photonic computing and quantum information processing.

Acknowledgments—This work is supported by the National Key Research and Development Program of China under Contract No. 2024YFA1610900 and the National Natural Science Foundation of China (NSFC) under Contract No. 12447106, No. 12541501, No. 12547102 and No. 12450404. D.D.H. also acknowledges the support of NSFC under No. 11875133 and No. 11075057, the National Key Research and Development Program of China under Contract No. 2018YFB2101302. The computations in this research were performed using the CFFF platform of Fudan University.

Data availability—The data that support the findings of this article are publicly available [43].

- [1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, England, 2010).

- [2] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, and G. J. Milburn, Linear optical quantum computing with photonic qubits, *Rev. Mod. Phys.* **79**, 135 (2007).
- [3] B. J. Shastri, A. N. Tait, T. Ferreira de Lima, W. H. Pernice, H. Bhaskaran, C. D. Wright, and P. R. Prucnal, Photonics for artificial intelligence and neuromorphic computing, *Nat. Photonics* **15**, 102 (2021).
- [4] P. L. McMahon, The physics of optical computing, *Nat. Rev. Phys.* **5**, 717 (2023).
- [5] E. Knill, R. Laflamme, and G. J. Milburn, A scheme for efficient quantum computation with linear optics, *Nature (London)* **409**, 46 (2001).
- [6] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund *et al.*, Deep learning with coherent nanophotonic circuits, *Nat. Photonics* **11**, 441 (2017).
- [7] D. Pérez, I. Gasulla, L. Cradginton, D. J. Thomson, A. Z. Khokhar, K. Li, W. Cao, G. Z. Mashanovich, and J. Capmany, Multipurpose silicon photonics signal processor core, *Nat. Commun.* **8**, 636 (2017).
- [8] X. Lin, Y. Rivenson, N. T. Yardimci, M. Veli, Y. Luo, M. Jarrahi, and A. Ozcan, All-optical machine learning using diffractive deep neural networks, *Science* **361**, 1004 (2018).
- [9] S. Pai, Z. Sun, T. W. Hughes, T. Park, B. Bartlett, I. A. Williamson, M. Minkov, M. Milanizadeh, N. Abebe, F. Morichetti *et al.*, Experimentally realized in situ back-propagation for deep learning in photonic neural networks, *Science* **380**, 398 (2023).
- [10] Z. Xue, T. Zhou, Z. Xu, S. Yu, Q. Dai, and L. Fang, Fully forward mode training for optical neural networks, *Nature (London)* **632**, 280 (2024).
- [11] T. Fu, J. Zhang, R. Sun, Y. Huang, W. Xu, S. Yang, Z. Zhu, and H. Chen, Optical neural networks: Progress and challenges, *Light Sci. Appl.* **13**, 263 (2024).
- [12] W. Bogaerts, D. Pérez, J. Capmany, D. A. Miller, J. Poon, D. Englund, F. Morichetti, and A. Melloni, Programmable photonic circuits, *Nature (London)* **586**, 207 (2020).
- [13] J. L. O'Brien, Optical quantum computing, *Science* **318**, 1567 (2007).
- [14] S. Konno, W. Asavanant, F. Hanamura, H. Nagayoshi, K. Fukui, A. Sakaguchi, R. Ide, F. China, M. Yabuno, S. Miki *et al.*, Logical states for fault-tolerant quantum computation with propagating light, *Science* **383**, 289 (2024).
- [15] PsiQuantum team, A manufacturable platform for photonic quantum computing, *Nature (London)* **641**, 876 (2025).
- [16] E. Pelucchi, G. Fagas, I. Aharonovich, D. Englund, E. Figueroa, Q. Gong, H. Hannes, J. Liu, C.-Y. Lu, N. Matsuda *et al.*, The potential and global outlook of integrated photonics for quantum technologies, *Nat. Rev. Phys.* **4**, 194 (2022).
- [17] I. Bente, S. Taheriniya, F. Lenzini, F. Brückhoff-Plückelmann, M. Kues, H. Bhaskaran, C. D. Wright, and W. Pernice, The potential of multidimensional photonic computing, *Nat. Rev. Phys.* **7**, 439 (2025).
- [18] S. Shekhar, W. Bogaerts, L. Chrostowski, J. E. Bowers, M. Hochberg, R. Soref, and B. J. Shastri, Roadmapping the next generation of silicon photonics, *Nat. Commun.* **15**, 751 (2024).
- [19] D. A. Miller, Self-configuring universal linear optical component, *Photonics Res.* **1**, 1 (2013).
- [20] J. Carolan, C. Harrold, C. Sparrow, E. Martín-López, N. J. Russell, J. W. Silverstone, P. J. Shadbolt, N. Matsuda, M. Oguma, M. Itoh *et al.*, Universal linear optics, *Science* **349**, 711 (2015).
- [21] G. Golub and W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, *J. Soc. Ind. Appl. Math.* **2**, 205 (1965).
- [22] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, Experimental realization of any discrete unitary operator, *Phys. Rev. Lett.* **73**, 58 (1994).
- [23] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, Optimal design for universal multiport interferometers, *Optica* **3**, 1460 (2016).
- [24] K. Ellis, C. Wong, M. Nye, M. Sablé-Meyer, L. Morales, L. Hewitt, L. Cary, A. Solar-Lezama, and J. B. Tenenbaum, Dreamcoder: Bootstrapping inductive program synthesis with wake-sleep library learning, in *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation* (Association for Computing Machinery (ACM), New York, 2021), pp. 835–850.
- [25] F. Mezzadri, How to generate random matrices from the classical compact groups, *Not. Am. Math. Soc.* **54**, 592 (2007), <https://www.ams.org/notices/200705/fea-mezzadri-web.pdf>.
- [26] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/49c3-4rp4> for details of the matrix-domain implementation, MZI primitive definitions, curriculum-learning procedure, representative discovered programs, structure-aware examples, and cross-dimensional generalization analysis.
- [27] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, The “wake-sleep” algorithm for unsupervised neural networks, *Science* **268**, 1158 (1995).
- [28] L. Sarra, K. Ellis, and F. Marquardt, Discovering quantum circuit components with program synthesis, *Mach. Learn. Sci. Technol.* **5**, 025029 (2024).
- [29] F. J. Ruiz, T. Laakkonen, J. Bausch, M. Balog, M. Berekatain, F. J. Heras, A. Novikov, N. Fitzpatrick, B. Romera-Paredes, J. Van De Wetering *et al.*, Quantum circuit optimization with alphasensor, *Nat. Mach. Intell.* **7**, 374 (2025).
- [30] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, Human-level concept learning through probabilistic program induction, *Science* **350**, 1332 (2015).
- [31] P. Liang, M. I. Jordan, and D. Klein, Learning programs: A hierarchical Bayesian approach, in *ICML* (Omnipress, Madison, 2010), Vol. 10, pp. 639–646.
- [32] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, Curriculum learning, in *Proceedings of the 26th annual international conference on machine learning* (Omnipress, Madison, 2009), pp. 41–48.
- [33] X. Wang, Y. Chen, and W. Zhu, A survey on curriculum learning, *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 4555 (2021).

- [34] G. H. Golub and C. F. Van Loan, *Matrix Computations* (JHU Press, Baltimore, Maryland, 2013).
- [35] P. A. Ivanov, E. Kyoseva, and N. V. Vitanov, Engineering of arbitrary $U(N)$ transformations by quantum householder reflections, *Phys. Rev. A* **74**, 022323 (2006).
- [36] C. Lu, X. Wang, and G. Ma, Experimental realization of special-unitary operations in classical mechanics by nonadiabatic evolutions, *Phys. Rev. Lett.* **135**, 027201 (2025).
- [37] H. M. Markowitz, The elimination form of the inverse and its application to linear programming, *Manage. Sci.* **3**, 255 (1957).
- [38] J. R. Gilbert, C. Moler, and R. Schreiber, Sparse matrices in matlab: Design and implementation, *SIAM J. Matrix Anal. Appl.* **13**, 333 (1992).
- [39] S. Lin, J. Zeng, S. Lin, S. Yu, and Y. Zhang, High-fidelity and compact topology architecture for large-scale reconfigurable linear optical networks, *Adv. Photonics Nexus* **4**, 066012 (2025).
- [40] K. H. R. Mojaver, B. Zhao, E. Leung, S. M. R. Safaei, and O. Liboiron-Ladouceur, Addressing the programming challenges of practical interferometric mesh based optical processors, *Opt. Express* **31**, 23851 (2023).
- [41] H. Zhou, H. Yang, N. Gangi, Z. R. Huang, H. Ren, and J. Gu, Apollo: Automated routing-informed placement for large-scale photonic integrated circuits, in *2025 IEEE/ACM International Conference On Computer Aided Design (ICCAD)* (IEEE, New York, 2025), pp. 1–9.
- [42] H. Zhou, K. Zhu, and J. Gu, Automated curvy waveguide routing for large-scale photonic integrated circuits, *arXiv*: 2410.01260.
- [43] Y. Zhang *et al.* (2026), [10.5281/zenodo.20523567](https://doi.org/10.5281/zenodo.20523567).