

Quantum element-wise transforms (extended abstract)

Zane M. Rossi¹ and Rahul Sarkar²

¹*Department of Physics, Graduate School of Science,*

The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033, Japan

²*Department of Mathematics, University of California, Berkeley, CA 94720, USA.*

Quantum algorithms for basic numerical linear algebraic tasks have proven essential for translating diverse problems to a unified quantum computational context. Many of these tasks—e.g., applying a function to the spectrum of a matrix embedded in a unitary process (a so-called *block encoding*)—are well-addressed by techniques like quantum singular value transformation (QSVT). However, there exist useful matrix transforms whose realization by quantum algorithms is unclear or inefficient. In this work we construct improved quantum algorithms for some of these transforms, the simplest of which is a *scalar function applied element-wise*. We show the auxiliary space required to compute *quantum element-wise transforms* can be reduced exponentially in the degree of the applied function compared to prior work, and raise and rectify errors in previous constructions. We present these algorithms alongside applications to machine learning, simulation, and signal processing.

I. Overview and main results

The field of classical algorithms relies on a rich sub-structure of numerical linear algebra [1, 2]. The success of this scheme is self-evident; matrix methods pervade signal processing, simulation, statistics, machine learning, and the discrete analogues of huge swaths of functional analysis [3].

Recently the field of quantum algorithms has warmed to a similar scheme. One manifestation is the increasing use of the *block encoding* input model, which embeds a matrix as a sub-block of a larger unitary process. The permissible transformations of block-encodings inform the complexities of most quantum algorithms inside a unified, numerically-stable framework [4, 5]. Crucially, for certain transformations and block encodings obeying constraints, these algorithms appear to prepare states and solve discrete problems far more efficiently than their classical cousins.

In this work we expand the quantum numerical linear algebraic toolkit to matrix transformations not easily captured by current block encoding algorithms. Concretely, **we provide quantum algorithms for computing element-wise functions of block encodings using exponentially less space in the degree of the applied function**, compared with prior work. These element-wise matrix functions appear ubiquitously in classical signal processing, machine learning, and statistics [6, 7], and consequently our work provides novel, highly space-efficient quantum algorithms for computing non-linear transformations of high-dimensional data.

The transformation we consider (Def. I.1) is distinct from that of QSVT, which applies a function to the singular values of a block-encoded matrix, or LCU, which takes linear combinations of block encodings. Our main theorem (Thm. I.1) provides the resource complexity of applying an arbitrary polynomial function *element-wise*, and yields an exponential reduction in space use compared with existing work (Tab. I). We refer to this (and related manipulations) as *quantum element-wise transforms* (QEWts) [2606.06456].

Method	Query comp.	Aux. space
Prior work [8]	d	$\mathcal{O}(ad + nd)$
Main result (Thm. II.3)	d	$\mathcal{O}(a + n(\log d))$
LCU variant (Thm. III.4)	d	$\mathcal{O}(a + n(\log d))$
Approx. variant (Cor. III.5.2)	$\mathcal{O}(d^{\log(\log(d/\varepsilon))})$	$\mathcal{O}(a + n \log(d^{\log(\log(d/\varepsilon))}))$
Half-compressed (Thm. C.1)	d	$\mathcal{O}(a + nd + \log d)$

TABLE I. Overview of the resource complexities of our main result, variants, and prior work. The problem considered is the element-wise application of a polynomial of degree d given an n -qubit matrix block-encoded using a -qubits of auxiliary space. Our main construction can be further space-compressed by an LCU-based variant, as well as amplified and approximated under certain input assumptions. The referenced theorems and corollaries provide detailed analysis and full circuits.

Definition I.1 (Element-wise function of a matrix). Let $M \in \mathbb{C}^{n \times m}$; then the element-wise application of $f : \mathbb{C} \rightarrow \mathbb{C}$ to M , denoted $f^\circ(M)$, is $a_{ij} \mapsto f(a_{ij})$ for a_{ij} the ij -th element of M in the computational basis.

Theorem I.1 (Quantum element-wise transform). Let f be a polynomial of degree d with no constant term satisfying $|f| \leq 1$ on $[-1, 1]$, with coefficients c_k . Then, given U an (α, a, ε) block encoding of an n -qubit operator A , there is a quantum circuit preparing a (β, b, δ) block encoding of $f^\circ(A/\alpha)$ —the *element-wise application of f to A* —requiring $\mathcal{O}(d)$ copies of controlled U and $U^\dagger$, and single/two-qubit gates, where (β, b, δ) satisfy

$$\beta = \sum_{k=1}^d |c_k|, \quad b = \mathcal{O}(a + n \log d), \quad \delta = \sum_{k=1}^d |c_k| \left[(1 + \varepsilon/\alpha)^k - 1 \right]. \quad (1)$$

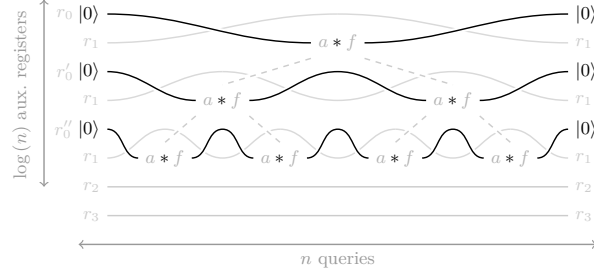


FIG. 1. Simplified depiction of our ‘weaving lemma’ (Lem. II.2). Here $(*)$ represents a quantum subroutine catalytically using copies of $|0\rangle_{r_0}$ by swapping the r_0 and r_1 registers. Catalysis here means that $(*)$ is block-diagonal with a known $(+1)$ -eigensubspace. The use by $(*)$ of r_0 is indicated by an allocation (a) and freeing (f). If each $(*)$ itself sequentially calls copies of $(*)$, then the total required copies of r_0 grows with the *depth* of the tree of recursive calls to $(*)$.

Beyond our space-compressed construction of QEWTs, we also provide a comprehensive treatment of prior work, **raising and rectifying errors which otherwise invalidate previous constructions**. This includes an efficient circuit for constructing so-called SELECT unitaries (Prop. C.2), which are required to apply linear combination of unitaries (LCU) [9–11] techniques to bootstrap element-wise powers to arbitrary element-wise functions. We identify and resolve errors in prior work’s SELECT constructions [8], allowing for the recovery of their main statements as well as a simple hybridization scheme to exchange circuit width and depth for QEWTs (Cor. II.3.1).

Finally, we devote Sec. III to a series of alternative methods for computing exact and approximate QEWTs. This includes a construction for further reducing the auxiliary space required by a constant amount (Thm. III.4), as well as a modification incorporating a variant of robust, oblivious amplitude amplification (ROAA) (Cor. III.5.2) to increase overall success probability for applying the desired block encoding.

II. Methods and prior work

The core of our construction depends on the orchestration of well-known and (to the authors’ knowledge) new but simple subroutines for manipulating block encodings. These include (1) a known identity relating the element-wise product of matrices to a principal submatrix of their Kronecker product, (2) a known construction for exponentially reducing the auxiliary space required when *matrix multiplying* many block encodings at once (the *compression gadget* of Low and Wiebe [12, 13]), and (3) a method for cheaply duplicating blocks of block encodings *which themselves have block structure* (which we call a *swap-copy*, Lem. II.1) *as well as* a way to efficiently re-use the auxiliary space required by the swap-copy inside recursively self-composed versions of our protocol, enabling space-efficient computation of high-degree element-wise products (our *weaving lemma*, Lem. II.2).

The primary contribution of our work, beyond constructing (3), is showing that (1), (2), and (3) interact while maintaining correctness and $\mathcal{O}(\log d)$ auxiliary space. Here we briefly describe (3) with associated Figs. 1 and 3.

Lemma II.1 (Swap-copy). Let U block encode $(A \oplus B)$, where $\dim B = (2^n - 1) \dim A$. Then using one query to U , a register of n qubits initialized to $|0^n\rangle$, and $2n$ swap gates, we can deterministically construct V block encoding $A^{\oplus 2^n}$. Moreover, after applying V , the additional register remains in the $|0^n\rangle$ state if the block encoding succeeds.

For expediency we only informally describe (along with Fig. 1) the action of the weaving lemma (Lem. II.2), which allows us to allocate a number of copies of the register r_0 used by the swap-copy operation that scales only with the *depth* of the tree of function-calls that recursively define our element-wise product (Figs. 2, 3). The core observation is that, when working with block encodings with known $(+1)$ -eigensubspaces, we re-use states in this subspace *across sequential applications* of this block encoding by recording each constituent block encoding’s successful application through a compression gadget.

To enforce block-diagonality of block encodings, which enables the catalytic re-use of auxiliary space by our nested swap-copy operations, we simultaneously rely on the well-known *compression gadget* [12] which records the success of serial block encoding multiplications in binary on an auxiliary register (denoted r_4 in Fig. 3). In each of our variant constructions, either by interstitial block encoding multiplications, or simple LCUs, we can, conditioned on the register r_4 ending up in the all zeros state, ensure block-diagonality of subroutines. This allows us to take a circuit for block encoding $(A \otimes B)$, which has as a principal sub-matrix $(A \circ B)$ (the element-wise product), and transform it into a block encoding of $I \otimes (A \circ B)$ on the *same registers*, permitting recursive self-composition.

To summarize, our construction first identifies our desired element-wise product with a matrix product (in this case a Kronecker product of block encodings), then introduces additional block encoding products to enforce block-diagonality, and then applies both (a) our swap-copy and (b) the standard compression gadget to a recursively

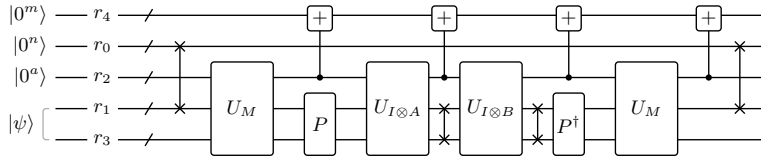


FIG. 2. The circuit for *clean element-wise products* of block encodings, i.e., block encodings of $I_{r_1} \otimes (A \circ B)_{r_3}$ using auxiliary space $r_2 r_4$. Note if $r_2 r_4$ is in the all-zeros state, then $|0^n\rangle$ in r_0 is used catalytically by the swap-copy. Here $(+)$ notation indicates projection-controlled addition to r_4 , used for the compression gadget. In our recursive construction each $U_{(I \otimes *)}$ is replaced by an analogous circuit which acquires (1) a new copy of r_0 for each recursive level, and (2) adds to r_4 .

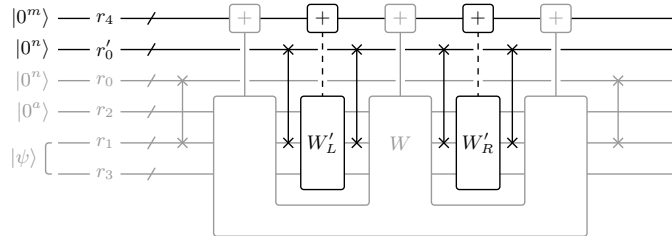


FIG. 3. A depiction of the first level of our recursive construction for taking element-wise powers of two block encodings W'_L, W'_R . Note the register used for the compression gadget, r_4 , is not duplicated, and remains the target for the addition gadgets of protocols deeper in the recursion. A new register of size n , r'_0 , is initialized to $|0\rangle$, and a new copy introduced at each level of recursion, though *re-used between swap-copies at the same level*. This catalysis succeeds under the condition that all block encodings are successfully applied (equivalently, that r_2 and r_4 end up in the all zeros state).

defined construction. Both (a) and (b) require space growing only logarithmically in the degree of the product. We then build an element-wise SELECT unitary, after which LCU techniques yield Thm. I.1.

Beyond the direct improvement to the linear-space QEWT of [8], our construction provides an operator extension to state-based non-linear transformation algorithms [14, 15], as well as a non-trivial extension to the widely applied compression gadget [12, 13]. By bringing the space, query, and gate complexity of our algorithm in line with other common matrix functions of block encodings, we enable the practical implementation of our construction on near-term devices, incorporating element-wise transforms, which appear widely in classical signal processing and machine learning, into the quantum algorithmic toolkit.

III. Discussion, applications, and impact

Our work constructs a new **quantum algorithm for computing element-wise functions of block encodings whose space use scales logarithmically rather than linearly in the degree of the applied function**—this scaling, matching that of common block encoding methods [4, 9], represents a novel and practical subroutine in the block encoding toolkit. Moreover, we rectify errors in previous work, establishing a robust foundation for the theory of *quantum element-wise transforms* (QEWTs).

Concretely, we devote Sec. IV of our work to specific applications across (1) machine learning inference, (2) multidimensional signal processing, and (3) generalized matrix products. For (1) we apply our main construction to reduce the space requirements for computing the element-wise softmax function of block encoding products (appearing in transformer inference), directly improving prior constructions [8, 16]. In (2) we show multidimensional analogues of the well-known convolution theorem allow us to space- and time-efficiently compute 2D convolutions of block encodings. Finally in (3) we provide quantum algorithms for well-known block-matrix products (the *Tracy-Singh* and *Khatri-Rao* products [17, 18]) whose algorithms follow from our main theorem.

Our work suggests numerous extensions, including the use of randomized/approximative techniques to further improve performance under architectural constraints [19–21]. We expect that additional knowledge of the input (e.g., low rank or well-conditioned) can be leveraged for performance, and that variants of our swap-copy and weaving lemma can yield independently interesting methods for manipulating block encodings with known block structure. Our aim is that quantum algorithms for core numerical linear algebraic subroutines can be efficiently and interpretably combined to (1) more easily incorporate sophisticated classical techniques and (2) admit the practical application of numerical tools for compilation/verification/optimization of quantum subroutines.

-
- [1] Lloyd N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2019.
 - [2] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra, Twenty-fifth Anniversary Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2022.
 - [3] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
 - [4] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2019.
 - [5] John M Martyn, Zane M Rossi, Andrew K Tan, and Isaac L Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2(4):040203, 2021.
 - [6] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.
 - [7] Grigorios G. Chrysos, Yongtao Wu, Razvan Pascanu, Philip H.S. Torr, and Volkan Cevher. Hadamard product in deep learning: Introduction, advances and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(8):6531–6549, 2025.
 - [8] Naixu Guo, Zhan Yu, Matthew Choi, Yizhan Han, Aman Agrawal, Kouhei Nakaji, Alán Aspuru-Guzik, and Patrick Rebentrost. Quantum transformer: Accelerating model inference via quantum linear algebra. *arXiv preprint, arXiv:2402.16714*, 2025.
 - [9] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Info. Comput.*, 12(11–12):901–924, Nov 2012.
 - [10] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse Hamiltonians. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing (STOC)*, STOC '14, page 283–292. ACM, May 2014.
 - [11] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 792–809. IEEE, 2015.
 - [12] Guang Hao Low and Nathan Wiebe. Hamiltonian simulation in the interaction picture. *arXiv preprint, arXiv:1805.00675*, 2019.
 - [13] Francisca Vasconcelos and András Gilyén. Methods for reducing ancilla-overhead in block encodings. *arXiv preprint, arXiv:2507.07900*, 2025.
 - [14] Naixu Guo, Kosuke Mitarai, and Keisuke Fujii. Nonlinear transformation of complex amplitudes via quantum singular value transformation. *Phys. Rev. Res.*, 6:043227, Dec 2024.
 - [15] Zoë Holmes, Nolan J. Coble, Andrew T. Sornborger, and Subaşı. Nonlinear transformations in quantum computation. *Phys. Rev. Res.*, 5:013105, Feb 2023.
 - [16] Arthur G. Rattew and Patrick Rebentrost. Non-linear transformations of quantum amplitudes: Exponential improvement, generalization, and applications. *arXiv preprint, arXiv:2309.09839*, 2023.
 - [17] Derrick S. Tracy and Rana P. Singh. A new matrix product and its applications in partitioned matrix differentiation. *Statistica Neerlandica*, 26(4):143–157, 1972.
 - [18] C. G. Khatri and C. Radhakrishna Rao. Solutions to some functional equations and their applications to characterization of probability distributions. *Sankhyā Ser. A*, 30(2):167–180, 1968.
 - [19] John M. Martyn and Patrick Rall. Halving the cost of quantum algorithms with randomization. *npj Quantum Inf.*, 11(47), 2025.
 - [20] Earl Campbell. Random compiler for fast Hamiltonian simulation. *Phys. Rev. Lett.*, 123(7), 2019.
 - [21] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Phys. Rev. X*, 11:011020, Feb 2021.