Quantum Algorithm for Clifford Multiplication

Kagwe A. Muchane[†]

Abstract

Given two dense multivectors of the Clifford algebra $\mathcal{C}\ell(V, Q)$ with $N = 2^{p+q}$ coefficients, the fastest known classical algorithms compute their geometric product in $O(N^{\omega/2})$ arithmetic operations, where ω denotes the matrix multiplication exponent. I show that, under amplitude encoding, a quantum computer executes the geometric product in $O(\text{polylog}\, N)$ time, using logarithmic space with sublogarithmic circuit depth. This exponential speedup establishes Clifford multiplication as a quantum primitive, providing an efficient computational foundation for quantum geometric algorithms and relativistic simulations.

Keywords: Clifford algebra; geometric product; quantum algorithms; quantum Fourier transform; multivectors; cocycle-twisted convolution.

AMS subject classifications: 15A66; 81P68; 68Q12; 68W10; 81R05.

1 Introduction

In his 1844 *Ausdehnungslehre* [12], Hermann Grassmann introduced the non-metric exterior product, later extending his theory to the central product, which incorporated metric notions of magnitude and projection [13]. William Kingdon Clifford then synthesized this dual product system with Hamilton's quaternions into a single associative engine by imposing the vector squaring relation $v^2 = Q(v)$, calling the resulting algebra *geometric algebra* [9]. By unifying the inner and exterior products into one associative multiplication, the geometric product has become a common language for Euclidean, projective, conformal, and spacetime geometry [11, 19], as well as 21st-century applications such as computer vision, robotics, computer graphics, and geometric machine learning [25, 4, 5]. Recent Clifford-based neural architectures replace conventional tensor operations by geometric products, allowing network layers to respect the geometric symmetries of the data [22, 23, 7, 6]. Likewise, Hestenes' spacetime algebra reformulates relativistic mechanics and field theory, where Lorentz transformations, spinors, and Minkowski spacetime are all expressed intuitively through Clifford multiplication [15, 16]. While broadly applicable, dense geometric algebra computations have proven to be computationally demanding. The difficulty is fundamentally combinatorial. Because multiplying two dense multivectors can mean combining exponentially many blade pairs, Clifford multiplication has represented one of the principal computational bottlenecks for large-scale geometric algebra computations.

The most powerful quantum algorithms demonstrate that, on a quantum computer, harmonic structure becomes computational advantage. The quantum Fourier transform [10] changes states to a basis in which that harmony becomes visible, forming the foundation of algorithms such as Shor's factoring algorithm [24] and hidden subgroup algorithms [17, 18]. In my most recent work, I discovered that Clifford's geometric product is not the sum of two independent products, but rather the harmonic decomposition of exchange under the action of a transposition $\tau \in S_2$. This replaces the traditional axiomatic view of the inner and exterior products as independent operations; they emerge as Fourier sectors of the exchange orbit of τ. That finding naturally raised the following question: *If the geometric product already possesses an intrinsic harmonic structure, can Clifford multiplication itself be realized as a quantum harmonic computation?*

The answer is *yes*. Admittedly, this is possible because the geometric product can be seen as cocycle-twisted convolution over the group $(\mathbb{Z}_2)^n$. I formulate quantum Clifford multiplication as a harmonic computation whose underlying group operation is separated from the orientation information carried by the cocycle. Unlike operator-based quantum treatments of Clifford algebras, which execute Clifford multiplication through the action of quantum operators on states [21], this algorithm computes the bilinear coefficient product without realizing the multivectors as operators. It computes dense Clifford multiplication with polylogarithmic gate complexity in the coefficient dimension, fitting neatly within the standard amplitude-encoding paradigm of quantum algorithms. The output is a quantum state representing the coefficient vector of the geometric

product, from which observables of the product may be efficiently estimated.

The historical perception of Clifford algebra as a specialized mathematical tool has persisted despite its conceptual and interpretational advantages. One contributing factor is that, until recently, those advantages had not been accompanied by a computational advantage. This work permanently changes that perspective; the circuit developed here is universal, and independent of the input multivectors. Efficient quantum multiplication of multivectors opens the possibility of quantum-native algorithms for geometric machine learning, Clifford-valued signal processing, spacetime and relativistic simulation, geometric robotics, and other computational problems whose natural language is geometric algebra.

2 Quantum Clifford Multiplication

Dense Clifford multiplication is the function problem of computing the geometric product

$$f : \mathcal{C}\ell_{p,q} \times \mathcal{C}\ell_{p,q} \to \mathcal{C}\ell_{p,q}, \qquad (A, B) \longmapsto AB.$$

Every element in $\mathcal{C}\ell_{p,q}$ is a linear combination $A = \sum_{x \in I} a_x e_x$, where I is the set of all $N = 2^n$ basis blades and $n = p + q$. Since $|I| = N$, each dense multivector is specified by N coefficients. By bilinearity of the geometric product,

$$f(A, B) = \sum_{x \in I} \sum_{y \in I} a_x b_y (e_x e_y). \tag{1}$$

There are two ways to measure the complexity of this problem. Treating the coefficient arrays themselves as the input, dense Clifford multiplication is polynomial in the input size $N = 2^n$ and therefore belongs to FP under the standard bit-complexity model. Treating the geometric dimension n as the scaling parameter exposes the combinatorial growth of the algebra. Since $N = 2^n$, every known classical algorithm for dense multivectors requires exponential time in n, placing the problem in $\mathsf{FEXPTIME}$. Throughout this paper we adopt the latter viewpoint, as the uniform, representation-independent setting for comparing classical and quantum algorithms on families of Clifford algebras. For a classical computer, the product of two dense multivectors requires evaluating one interaction for every ordered pair of basis blades. Since an n-dimensional Clifford algebra contains $N = 2^n$ basis blades, the straightforward coordinate expansion performs

$$\Theta(N^2) = \Theta(4^n)$$

blade interactions.

Modern classical improvements focus on reducing the cost of individual blade interactions or exploiting alternative representations. Basis blades are encoded as bitstrings, output blades are computed by bitwise XOR, sign is evaluated using parity operations, and SIMD or GPU implementations evaluate many blade products simultaneously [1, 11]. These optimizations significantly improve constant factors and practical performance, but they do not alter the

asymptotic complexity of dense multiplication, as every nonzero coefficient of the first multivector may still interact with every nonzero coefficient of the second. Consequently, dense Clifford multiplication retains quadratic complexity in the coefficient dimension, or equivalently, exponential complexity in the geometric dimension n. The fastest known asymptotic algorithms instead exploit faithful matrix representations of finite-dimensional Clifford algebras, thereby reducing dense Clifford multiplication to matrix multiplication. The best current classical matrix multiplication algorithms run in time $O\big(N^{\omega/2}\big)$, with the current optimal upper bound being $\omega < 2.371339$ [3] for the matrix multiplication exponent. This improves the dependence on the coefficient dimension from quadratic to $O(N^{1.186})$, but the computation remains exponential in the geometric dimension,

$$O\Big((2^n)^{\omega/2}\Big) = O\Big(2^{(\omega/2)n}\Big).$$

The quantum algorithm presented in this paper follows a different approach, exploiting the algebraic structure of the geometric product itself. I present an efficient quantum circuit whose complexity scales linearly with the geometric dimension, with the precise gate and depth bounds established in Theorem 2.5. To expose the algebra responsible for this speedup, we first examine the multiplication law of the basis blades. Their binary encoding separates Clifford multiplication into a group operation on blade indices together with a cocycle encoding orientation.

Basis encoding. Every basis blade is uniquely determined by a subset of the basis vectors and is therefore identified with its characteristic bit vector. Writing

$$x = \sum_{i=1}^{n} x_i 2^{\,i-1}, \qquad x_i \in \{0,1\},$$

we identify the integer bitmask x with the blade

$$e_x = e_1^{x_1} e_2^{x_2} \cdots e_n^{x_n}.$$

Throughout, bitstrings are displayed in the conventional most-significant-bit-first order $(x_n, \ldots, x_1)$. The displayed bitstring records the presence of generators $(e_n, \ldots, e_1)$, while the integer encoding is little-endian.

Table 1: Binary encoding of basis blades together with representative products in $\mathcal{C}\ell_{3,0}(\mathbb{R})$. The output blade is determined by bitwise XOR, while the Clifford cocycle contributes the orientation sign.

e_x	x	e_y	y	$x \oplus y$	$\chi_{3,0}(x,y)$	$e_x e_y$
1	000	1	000	000	$+1$	1
e_1	001	e_2	010	011	$+1$	e_{12}
e_2	010	e_{13}	101	111	-1	$-e_{123}$
e_{12}	011	e_{123}	111	100	-1	$-e_3$
e_3	100	e_1	001	101	-1	$-e_{13}$
e_{13}	101	e_{12}	011	110	$+1$	e_{23}
e_{23}	110	e_3	100	010	$+1$	e_2
e_{123}	111	e_{23}	110	001	-1	$-e_1$

The geometric product of basis blades separates into an output index and a phase:

$$e_x e_y = \chi_{p,q}(x,y) e_{x\oplus y},$$

where $\oplus$ denotes bitwise XOR and $\chi_{p,q}(x,y) \in \{\pm 1\}$ is the Clifford cocycle. The XOR operation determines the output blade, while the Clifford cocycle encodes the orientation information from basis exchanges together with the metric signs determined by the signature (p,q).

Twisted Convolution

In recent work on generalized Clifford geometry [20], I showed that the geometric product can be understood through a harmonic decomposition under the exchange action of transposition τ. Though the full theory is developed elsewhere, it suffices to note that the $\mathbb{Z}_2$ Fourier transform diagonalizes the exchange operator τ, whose spectral projectors $\frac{1}{2}(1 \pm \tau)$ separate an ordered product ab into its symmetric and antisymmetric exchange sectors. For two-vectors, the exchange orbit is $(ab, \tau(ab)) = (ab, ba)$, and the normalized character table of $\mathbb{C}[\mathbb{Z}_2]$ acts by

$$F_2 \begin{pmatrix} ab \\ \tau(ab) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} ab \\ ba \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} ab + ba \\ ab - ba \end{pmatrix} = \sqrt{2} \begin{pmatrix} a \cdot b \\ a \wedge b \end{pmatrix}.$$

Thus, up to normalization, the orthogonal Clifford decomposition is the $\mathbb{Z}_2$ Fourier decomposition of the exchange orbit. The same exchange decomposition naturally extends from vectors to arbitrary basis blades and, by linearity, to dense multivectors.

For basis blades, every product $e_x e_y$ determines an exchange orbit $(e_x e_y, e_y e_x)$, whose relative orientation is encoded by the Clifford cocycle. The exchange rotor introduced in [20] is defined by $R = (ba)^{-1} ab$. For basis blades $a = e_x$ and $b = e_y$, the geometric product is

$$ab = \chi_{p,q}(x,y) e_{x\oplus y}, \qquad ba = \chi_{p,q}(y,x) e_{x\oplus y}.$$

Both orderings produce the same output blade $e_{x\oplus y}$, differing only in their cocycle values. Since every nonzero basis blade is invertible,

$$\begin{aligned} R &= (ba)^{-1}ab \\ &= \left(\chi_{p,q}(y,x)e_{x\oplus y}\right)^{-1}\left(\chi_{p,q}(x,y)e_{x\oplus y}\right) \\ &= \frac{\chi_{p,q}(x,y)}{\chi_{p,q}(y,x)}e_{x\oplus y}^{-1}e_{x\oplus y} \\ &= \frac{\chi_{p,q}(x,y)}{\chi_{p,q}(y,x)}. \end{aligned}$$

The Clifford cocycle therefore completely determines the multiplication law of the basis blades. By contrast, the exchange rotor measures only the relative effect of reversing the order of multiplication, which is the ratio between the two cocycle values. This distinction becomes clearer by decomposing the cocycle into exchange and metric contributions. The sign in Clifford multiplication has two independent contributions: the parity of transpositions required to reorder basis vectors, and the signature-dependent signs from the contractions $e_i^2 = \pm 1$. Since these contributions occur independently during multiplication, the cocycle factors as

$$\chi_{p,q}(x,y) = \chi_{\text{ex}}(x,y)\chi_{\text{met}}(x,y).$$

χ_{ex} records the signs introduced when restoring canonical blade order, while χ_{met} records the metric signs from repeated generators. Since the metric contribution is symmetric and $\chi_{\text{met}}(x,y) = \chi_{\text{met}}(y,x)$, it cancels in the ratio, giving

$$R = \frac{\chi_{\text{ex}}(x,y)}{\chi_{\text{ex}}(y,x)}.$$

The exchange rotor depends only on the exchange (orientation) component of the cocycle.

By linearity, dense Clifford multiplication is a superposition of such decorated exchange orbits. For dense multivectors $A = \sum_x a_x e_x$ and $B = \sum_y b_y e_y$,

$$AB = \sum_{x,y} a_x b_y \chi_{p,q}(x,y) e_{x\oplus y}.$$

Grouping together those having the same output blade $z = x \oplus y$ gives

$$AB = \sum_z c_z e_z, \qquad c_z = \sum_{x\oplus y=z} a_x b_y \chi_{p,q}(x,y) = \sum_x a_x b_{x\oplus z}\chi_{p,q}(x, x\oplus z).$$

The coefficient map $c_z = \sum_{x\oplus y=z} a_x b_y \chi_{p,q}(x,y)$ is a cocycle-twisted convolution over $(\mathbb{Z}_2)^n$ [2, 19]. For a Clifford algebra $\mathcal{C}\ell_{p,q}(\mathbb{K})$, with $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$, one has

$$\mathcal{C}\ell_{p,q}(\mathbb{K}) \cong \mathbb{K}_\chi[(\mathbb{Z}_2)^n], \qquad n = p+q,$$

where the cocycle χ is determined by the quadratic form Q after choosing an orthogonal basis. Ordinary convolution on the Abelian group $(\mathbb{Z}_2)^n$ is

diagonalized by the Walsh–Hadamard transform, the character table of $(\mathbb{Z}_2)^n$. Clifford multiplication differs only by the additional cocycle factor

$$(f *_\chi g)(z) = \sum_x f(x) g(x \oplus z) \chi(x, x \oplus z).$$

Passing to complex scalars does not alter the underlying twisted group algebra.

$$\mathbb{C} \otimes_{\mathbb{R}} \mathcal{C}\ell(V, Q) \cong \mathbb{C}_\chi[(\mathbb{Z}_2)^n],$$

so the blade-index group, cocycle, and twisted convolution are unchanged; only the coefficient field is extended. Since quantum amplitudes are already complex, the quantum circuit, including the cocycle phase oracle, covers complexified Clifford algebra without changing the asymptotic resource bounds. This reformulation is the key to the quantum algorithm. Rather than evaluating all 4^n blade interactions individually, the circuit computes the cocycle-twisted convolution coherently in amplitude space by combining reversible XOR, a diagonal cocycle phase oracle, and a Walsh–Hadamard transform on the summation index.

2.1 Algorithm

The input to the algorithm is two amplitude-encoded multivectors, where each computational basis state $|x\rangle$ corresponds to the Clifford basis blade e_x. The algorithm is executed by a universal quantum circuit $\mathcal{U}$ that prepares an amplitude encoding of the geometric product $AB = \sum_z c_z e_z$, that is,

$$\mathcal{U} : \left(|A\rangle = \sum_x a_x |x\rangle,\ |B\rangle = \sum_y b_y |y\rangle \right) \longmapsto |AB\rangle = \sum_z c_z |z\rangle, \tag{2}$$

where the coefficients $\{c_z\}$ are those of the geometric product AB. Because the output remains amplitude encoded, the multiplication primitive can be applied repeatedly, and chains of geometric products may be evaluated coherently within a larger quantum computation before a final measurement is performed.

The cocycle oracle

Having reduced Clifford multiplication to cocycle-twisted convolution, whose group operation is bitwise XOR, the remaining challenge is the implementation of the cocycle. To this end, I design a quantum oracle U_χ for its phase. For $\mathcal{C}\ell_{p,q}$, the Clifford cocycle is encoded by the Boolean phase polynomial

$$\Phi_{p,q} : (\mathbb{Z}_2)^n \times (\mathbb{Z}_2)^n \longrightarrow \mathbb{Z}_2,$$

which records the Boolean exponent of the cocycle. With the little-endian blade encoding fixed above, $e_x e_y = \chi_{p,q}(x, y) e_{x \oplus y}$, where

$$\chi_{p,q}(x, y) = (-1)^{\Phi_{p,q}(x,y)}$$

and

$$\Phi_{p,q}(x,y) = \sum_{1\le i<j\le n} x_j y_i + \sum_{i=p+1}^{n} x_i y_i \pmod 2.$$

The phase polynomial separates into two independent contributions. The first term records the parity of the exchanges needed to restore canonical blade order: a generator e_j from the left factor crosses a generator e_i from the right factor whenever $i < j$. The second term records the metric contribution from repeated generators in the negative directions, where $e_i^2 = -1$. Thus the exchange term is purely combinatorial, depending only on the ordering of basis generators, while the metric term depends only on the signature (p, q) of the defining quadratic form. Since

$$\Phi_{p,q} = \Phi_{\mathrm{ex}} + \Phi_{\mathrm{met}} \pmod 2,$$

the Clifford cocycle factorizes as $\chi_{p,q}(x,y) = \chi_{\mathrm{ex}}(x,y)\chi_{\mathrm{met}}(x,y)$, where

$$\chi_{\mathrm{ex}}(x,y) = (-1)^{\Phi_{\mathrm{ex}}(x,y)}, \qquad \chi_{\mathrm{met}}(x,y) = (-1)^{\Phi_{\mathrm{met}}(x,y)}.$$

The oracle mirrors this decomposition by first computing the Boolean exponent of the exchange cocycle through a reversible linear transformation, then using phase kickback to implement the map $\Phi_{\mathrm{ex}}(x,y) \longrightarrow (-1)^{\Phi_{\mathrm{ex}}(x,y)}$. An additional diagonal phase layer contributes the metric phase, so that

$$\chi_{p,q}(x,y) = \chi_{\mathrm{ex}}(x,y)\chi_{\mathrm{met}}(x,y) = (-1)^{\Phi_{\mathrm{ex}}(x,y)+\Phi_{\mathrm{met}}(x,y)} = (-1)^{\Phi_{p,q}(x,y)}.$$

A direct implementation of the exchange phase would apply a controlled-Z gate for every pair (i, j) with $i < j$. Since there are $\binom{n}{2}$ such pairs, this naive oracle uses $O(n^2)$ two-qubit gates to prepare $(-1)^{\sum_{i<j} x_j y_i}$. The metric contribution is diagonal and requires only q controlled-Z gates. Although this already reduces the dependence from the exponential coefficient dimension to the geometric dimension, the exchange phase order can be synthesized substantially more efficiently.

Lemma 2.1 (Nilpotent shift representation of the exchange polynomial)**.** *Let $S \in M_n(\mathbb{F}_2)$ be the lower nilpotent shift matrix,*

$$S = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \qquad S^n = 0.$$

Then the exchange contribution to the Boolean phase polynomial satisfies

$$\sum_{1\le i<j\le n} x_j y_i = x^T L y,$$

where

$$L = \sum_{k=1}^{n-1} S^k.$$

Proof. For each $k \geq 1$, the matrix S^k has ones on the k-th subdiagonal.

$$L = \sum_{k=1}^{n-1} S^k = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix},$$

so

$$x^T L y = \sum_{1 \leq i < j \leq n} x_j y_i.$$

The bilinear form $x^T L y$ therefore computes the exchange contribution $\Phi_{\text{ex}}(x, y)$ to the Boolean phase polynomial. Now define $\mathbf{T} := (I + S)^{-1}$. Since $(I + S)(I + S + \cdots + S^{n-1}) = I$, we have

$$\mathbf{T} = I + S + \cdots + S^{n-1} = I + L.$$

Unlike an infinite Neumann series, this resolvent expansion terminates exactly because S is nilpotent. Moreover, since subtraction equals addition in $\mathbb{F}_2$,

$$L = \mathbf{T} - I = \mathbf{T} + I.$$

□

Remark. $\mathbf{T}$ has three equivalent interpretations:

(i) *the finite resolvent of the nilpotent shift operator*,
(ii) *a lower-triangular Toeplitz matrix over* $\mathbb{F}_2$, and
(iii) *the binary prefix-sum transform.*

Viewed as the binary prefix-sum transform, $\mathbf{T}$ is an accumulation operator. The shift S advances one position through the ordered blade indices, while the finite series $I + S + S^2 + \cdots + S^{n-1}$ accumulates the exchange contributions required to restore canonical blade order. The resolvent form of $\mathbf{T}$ admits two complementary factorizations. The first is an elementary factorization into nearest-neighbor transvections. The second is a dyadic factorization into powers of the nilpotent shift.

Lemma 2.2 (Elementary sweep factorization)**.** *Let* $e_{i+1,i} \in M_n(\mathbb{F}_2)$ *denote the matrix with a single nonzero entry in position* $(i + 1, i)$*, and define*

$$E_{i+1,i} := I + e_{i+1,i}.$$

Then

$$\mathbf{T} = E_{n,n-1} E_{n-1,n-2} \cdots E_{2,1}.$$

Proof. The elementary matrix $E_{i+1,i}$ acts on column vectors by adding coordinate i into coordinate $i + 1$,

$$y_{i+1} \longmapsto y_{i+1} + y_i,$$

leaving all other coordinates fixed. Since the product acts on column vectors from right to left, the factorization applies the sweep

$$1 \to 2, \quad 2 \to 3, \quad \ldots, \quad n-1 \to n.$$

After this sweep, the j-th output coordinate is $y_1 + y_2 + \cdots + y_j$. Therefore the product has entries

$$\left(E_{n,n-1} \cdots E_{2,1}\right)_{ji} = \begin{cases} 1, & i \le j, \\ 0, & i > j, \end{cases}$$

which is exactly the unit lower-triangular matrix $\mathbf{T}$. □

Lemma 2.3 (Dyadic factorization)**.** *Let $h = \lceil \log_2 n \rceil$. Then*

$$\mathbf{T} = \prod_{m=0}^{h-1} \left(I + S^{2^m}\right).$$

Proof. First suppose $n = 2^r$. Define $P_r := \prod_{m=0}^{r-1} \left(I + S^{2^m}\right)$. We prove by induction that $P_r = I + S + S^2 + \cdots + S^{2^r-1}$. For $r = 1$, this is immediate. If the identity holds for r, then

$$\begin{aligned} P_{r+1} &= P_r \left(I + S^{2^r}\right) \\ &= \left(\sum_{k=0}^{2^r-1} S^k\right) + \left(\sum_{k=0}^{2^r-1} S^{k+2^r}\right) \\ &= \sum_{k=0}^{2^{r+1}-1} S^k. \end{aligned}$$

For general n, take $h = \lceil \log_2 n \rceil$. Since $S^n = 0$, all terms S^k with $k \ge n$ vanish, so the identity truncates to

$$\prod_{m=0}^{h-1} \left(I + S^{2^m}\right) = \sum_{k=0}^{n-1} S^k = \mathbf{T}.$$

□

Proposition 2.4 (Optimized Clifford cocycle oracle)**.** *The Clifford cocycle phase oracle*

$$U_\chi : |x\rangle\,|y\rangle \longmapsto (-1)^{\Phi_{p,q}(x,y)}\,|x\rangle\,|y\rangle$$

can be synthesized in two ways. The elementary sweep uses $O(n)$ two-qubit gates and depth $O(n)$. The dyadic method uses $O(n \log n)$ two-qubit gates and depth $O(\log n)$.

Proof. Both constructions follow from factorizations of the nilpotent resolvent $\mathbf{T} = (I+S)^{-1}$. Since $L = \mathbf{T}+I$ over $\mathbb{F}_2$, we have $x^T L y = x^T \mathbf{T} y + x^T y$. Therefore

$$\Phi_{p,q}(x,y) = x^T \mathbf{T} y + x^T y + \sum_{i=p+1}^{n} x_i y_i.$$

The last two terms combine over $\mathbb{F}_2$,

$$x^T y + \sum_{i=p+1}^{n} x_i y_i = \sum_{i=1}^{p} x_i y_i.$$

Thus the nontrivial exchange contribution is implemented through

$$U_{\mathbf{T}} : |y\rangle \longmapsto |\mathbf{T}y\rangle .$$

Since $\mathbf{T} \in GL_n(\mathbb{F}_2)$, the map $y \mapsto \mathbf{T}y$ is a bijection of $(\mathbb{Z}_2)^n$, and $U_{\mathbf{T}} |y\rangle = |\mathbf{T}y\rangle$ is a unitary permutation of the computational basis. A parallel controlled-Z layer between corresponding qubits of $|x\rangle$ and $|\mathbf{T}y\rangle$ kicks back the phase $(-1)^{x^T \mathbf{T} y}$, since the i-th controlled-Z gate contributes $(-1)^{x_i (\mathbf{T}y)_i}$, and the phases multiply to

$$(-1)^{\sum_i x_i (\mathbf{T}y)_i} = (-1)^{x^T \mathbf{T} y}.$$

Applying $U_{\mathbf{T}}^{\dagger}$ restores the second register to $|y\rangle$ while preserving this phase. A final controlled-Z layer over the positive directions contributes $(-1)^{\sum_{i=1}^{p} x_i y_i}$. Using $L = \mathbf{T} + I$ and the fact that addition and subtraction coincide over $\mathbb{F}_2$, we may rewrite

$$x^T \mathbf{T} y = x^T (\mathbf{T} + I) y + x^T y.$$

Hence the total phase is

$$\begin{aligned}
(-1)^{x^T \mathbf{T} y + \sum_{i=1}^{p} x_i y_i} &= (-1)^{x^T (\mathbf{T}+I) y + x^T y + \sum_{i=1}^{p} x_i y_i} \\
&= (-1)^{x^T (\mathbf{T}+I) y + \sum_{i=p+1}^{n} x_i y_i} \\
&= (-1)^{\Phi_{p,q}(x,y)}.
\end{aligned}$$

□

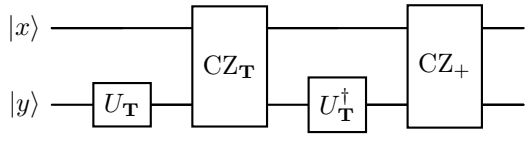

Figure 1: Optimized Clifford cocycle oracle U_χ. The transform $U_{\mathbf{T}}$ maps $|y\rangle$ to $|\mathbf{T}y\rangle$. The layer $\mathrm{CZ}_{\mathbf{T}}$ produces the phase $(-1)^{x^T \mathbf{T} y}$. The inverse transform uncomputes the second register, and CZ_+ contributes the remaining diagonal phase over the positive directions.

Implementation	Gates	Qubits	Depth
Naive	$O(\log^2 N)$	$O(\log N)$	$O(\log^2 N)$
Sweep	$O(\log N)$	$O(\log N)$	$O(\log N)$
Dyadic prefix	$O(\log N \log\log N)$	$O(\log N)$	$O(\log\log N)$

Table 2: Circuit resources for three implementations of the Clifford cocycle phase oracle U_χ, where $N = 2^n$ is the number of Clifford basis blades. Gate count measures elementary Clifford gates, qubit count measures the two input registers and any working space, and depth assumes parallel execution of disjoint gates.

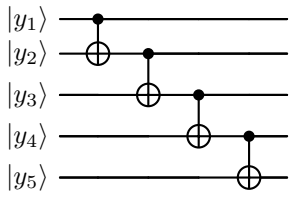

Figure 2: Elementary sweep variant of $U_{\mathbf{T}}$ for $n = 5$. Each CNOT implements the transvection $E_{i+1,i}$, propagating the binary prefix sum through the register. The circuit maps $|y\rangle \mapsto |\mathbf{T}y\rangle$ with $n-1$ CNOT gates and depth $n-1$.

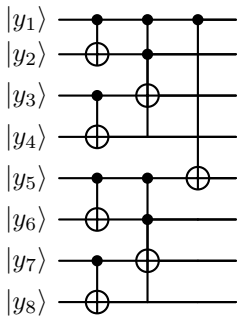

Figure 3: Schematic dyadic prefix construction for $U_{\mathbf{T}}$. The m-th stage propagates partial prefix sums across distance 2^m. With a parallel-prefix implementation, the propagation distance doubles at each stage, giving depth $O(\log n)$ and $O(n \log n)$ CNOT gates.

For the sweep, $U_{\mathbf{T}}$ uses $n-1$ CNOT gates, and $U_{\mathbf{T}}^\dagger$ uses another $n-1$. The central $\mathrm{CZ}_{\mathbf{T}}$ layer consists of n controlled-Z gates, and the final positive-direction diagonal layer consists of p controlled-Z gates. Hence the sweep oracle uses

$$(n-1) + (n-1) + n + p = 3n - 2 + p$$

two-qubit gates. Since $0 \le p \le n$, this is between $3n-2$ and $4n-2$, so both its gate count and depth are $O(n)$. The dyadic variant requires $O(n \log n)$

two-qubit gates and depth $O(\log n)$. The sweep minimizes gate count, while the dyadic method minimizes circuit depth. Since the geometric dimension satisfies $n = \log_2 N$, where N is the number of Clifford basis blades, the dyadic oracle has sublogarithmic depth $O(\log n) = O(\log \log N)$, and gate complexity $O(n \log n) = O(\log N \, \log \log N)$.

Procedure.

The following section combines the oracle with reversible XOR and the Walsh–Hadamard transform to complete the algorithm for Clifford multiplication.

Theorem 2.5. *Let $A, B \in \mathcal{C}\ell_{p,q}(\mathbb{K})$, with coefficient states $|A\rangle = \sum_x a_x|x\rangle$ and $|B\rangle = \sum_y b_y|y\rangle$. There exists an $O(n)$ circuit whose trivial-character branch prepares the normalized state proportional to $\sum_z c_z|z\rangle$. The success probability of this branch is $p_0 = 2^{-n}\sum_z |c_z|^2$.*

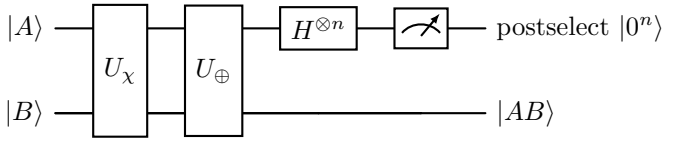

Figure 4: Quantum Clifford multiplication circuit.

Proof. Starting from the tensor-product state $|A\rangle|B\rangle = \sum_{x,y} a_x b_y |x\rangle \otimes |y\rangle$, apply the cocycle phase oracle to the two registers

$$U_\chi : |x\rangle|y\rangle \longmapsto (-1)^{\Phi_{p,q}(x,y)}|x\rangle|y\rangle.$$

The oracle depends only on the Clifford multiplication law and is independent of the coefficients of A and B. Next apply the reversible XOR map

$$U_\oplus : |x\rangle|y\rangle \longmapsto |x\rangle|x \oplus y\rangle,$$

which computes the underlying group operation on blade indices. Writing $z = x \oplus y$, equivalently $y = x \oplus z$, the state becomes,

$$|\psi\rangle = \sum_{x,z} a_x b_{x\oplus z} \chi_{p,q}(x, x \oplus z)|x\rangle|z\rangle.$$

Now apply the Walsh-Hadamard transform $H^{\otimes n}$ to the first register (the A-register).

$$H^{\otimes n}|x\rangle = 2^{-n/2} \sum_{w \in (\mathbb{Z}_2)^n} (-1)^{w\cdot x}|w\rangle,$$

we obtain

$$|\psi\rangle = 2^{-n/2} \sum_{w,z} \left(\sum_x (-1)^{w\cdot x} a_x b_{x\oplus z} \chi_{p,q}(x, x \oplus z) \right) |w\rangle|z\rangle.$$

The trivial-character branch $w = 0^n$ is therefore

$$2^{-n/2}|0^n\rangle \sum_z \left(\sum_x a_x b_{x\oplus z} \chi_{p,q}(x, x \oplus z) \right) |z\rangle = 2^{-n/2}|0^n\rangle \sum_z c_z |z\rangle,$$

where the inner sum is exactly the z-th coefficient of the Clifford product AB. Conditioned on this outcome, the second register is the normalized amplitude encoding of the geometric product AB. The success probability is the squared norm of the trivial-character branch,

$$p_0 = \left\| 2^{-n/2} \sum_z c_z |z\rangle \right\|^2 = 2^{-n} \sum_z |c_z|^2.$$

The reversible XOR uses $O(n)$ CNOT gates, and the Walsh–Hadamard transform uses n Hadamard gates. Thus the circuit has size

$$O(n) + \mathrm{cost}(U_\chi).$$

Using the linear-size sweep cocycle oracle gives an $O(n)$-gate circuit; using the dyadic oracle gives $O(n \log n)$. □

Remark (Twisted group algebra form). The proof of Theorem 2.5 depends only on four primitives:

(i) a basis $\{e_g : g \in \mathcal{G}\}$ indexed by a finite group,

(ii) an efficient reversible implementation of the group operation,

(iii) a twisted multiplication $e_g e_h = \chi_{\mathcal{G}}(g, h) e_{gh}$,

(iv) a quantum Fourier transform for $\mathcal{G}$, enabling postselection onto the trivial representation.

The proof is therefore independent of the specific Clifford algebra and should apply verbatim to any efficiently presented twisted group algebra $\mathbb{C}_\chi[\mathcal{G}]$. Let $\chi = \chi_{\mathcal{G}}$ be a normalized $U(1)$-valued 2-cocycle on $\mathcal{G}$. For $A = \sum_g a_g e_g, B = \sum_h b_h e_h$, the product coefficients are $c_k = \sum_{g\in\mathcal{G}} a_g b_{g^{-1}k} \chi_{\mathcal{G}}(g, g^{-1}k)$. After applying the cocycle phase, the reversible multiplication map $|g\rangle|h\rangle \mapsto |g\rangle|gh\rangle$, and the quantum Fourier transform on the first register, postselection on the trivial representation prepares $|\mathcal{G}|^{-1/2} \sum_k c_k |k\rangle$. The Clifford case is $\mathcal{G} = (\mathbb{Z}_2)^n$, where $gh = g \oplus h$ and the Fourier transform is $H^{\otimes n}$. In general, the cost per attempt is driven by the reversible group multiplication, the cocycle phase oracle, and the quantum Fourier transform over $\mathcal{G}$, with any additional overhead determined by the success probability of the trivial-representation branch.

This method differs from recent quantum algorithms for group convolution [8], which usually treat convolution as a linear operator determined by a fixed filter. Here the circuit carries out the bilinear multiplication of a twisted group algebra. The cocycle is incorporated explicitly through the diagonal phase oracle U_χ, while the Walsh–Hadamard transform performs the harmonic summation over the blade-index group.

Conjecture (Cocycle degree and Clifford hierarchy)**.** *Let $\mathcal{G}$ be a finite abelian group, and let*

$$\chi_{\mathcal{G}}(g,h) = \omega^{\Phi(g,h)}$$

be a phase cocycle whose exponent $\Phi : \mathcal{G}\times\mathcal{G} \to \mathbb{Z}_m$ has a polynomial representation of degree d in finite abelian coordinates for g and h. Then the cocycle phase oracle

$$U_{\chi,\mathcal{G}} : |g\rangle\,|h\rangle \longmapsto \chi_{\mathcal{G}}(g,h)\,|g\rangle\,|h\rangle$$

can be synthesized using gates from the dth level of the Clifford hierarchy.

The cocycle exponent $\Phi_{p,q}(x,y)$ from Clifford multiplication is quadratic in the blade coordinates, and explains why the phase oracle lives entirely within $\mathcal{C}_2$. This is the motivating example for the conjecture, which asserts that the algebraic degree of the cocycle exponent influences the Clifford-hierarchy level needed to implement the associated diagonal phase oracle. If true, the conjecture would provide a systematic design principle for quantum phase oracles of twisted group algebras, thus offering a guide for the construction of efficient multiplication circuits and symmetry-based quantum algorithms beyond the Clifford case. A possible approach is to decompose the cocycle exponent into coordinate monomials and synthesize the diagonal phase gates individually. If every degree-d monomial phase belongs to the dth level of the Clifford hierarchy, then the conjecture follows from the commutativity of diagonal phase operators.

2.2 Analysis

The complexity results established above and below concern the quantum model of computation. As is the case with many quantum linear algebra algorithms, recovering all $N = 2^n$ output coefficients classically requires $\Omega(N)$ measurements independent of the circuit complexity, an overhead not specific to Clifford multiplication but reflecting the general cost of extracting exponentially many classical values from a quantum state. The algorithm is best for computations in which the inputs are already available as quantum states or the output is consumed by subsequent quantum operations, expectation-value estimation, or measurement.

State preparation presents a complementary challenge, as preparing an arbitrary amplitude-encoded state generally requires $\Omega(N)$ resources in the worst case. The relevant question is not whether arbitrary multivectors can be prepared efficiently, but whether Clifford multivectors can be classified in such a way that they have efficient quantum representations. Several important families already do. Basis blades require only computational-basis preparation, the uniform multivector is obtained by applying $H^{\otimes n}$ to $|0^n\rangle$, stabilizer states prepared by Clifford circuits define structured amplitude-encoded multivectors in the blade basis, and tensor-product states give tensor-product multivectors. Identifying the broader class of efficiently preparable Clifford multivectors remains an open problem.

Measurement

The circuit prepares the desired product state in the trivial-character branch $N^{-1/2}|0^n\rangle \sum_z c_z|z\rangle$. Therefore

$$p_0 = N^{-1} \sum_z |c_z|^2 = 2^{-n}\|AB\|_2^2.$$

If the output coefficient vector is normalized, then $p_0 = 2^{-n}$. Normalized output does not imply large success probability; it gives the generic exponential postselection penalty. Since $c_z = \sum_x a_x b_{x\oplus z}\chi(x, x\oplus z)$, Young's inequality gives

$$\|AB\|_2 \le \|A\|_1\|B\|_2 \le 2^{n/2}\|A\|_2\|B\|_2.$$

For normalized inputs, this results in the universal bound $p_0 \le 1$, with equality only if $\|AB\|_2^2 = 2^n$.

This estimate alone does not characterize the typical success probability. For random normalized dense coefficient vectors, the sums defining c_z behave heuristically like random walks, suggesting $\|AB\|_2^2 = O(1)$ and $p_0 = O(2^{-n})$. Generic dense inputs are therefore expected to represent the least favorable regime for both classical and quantum Clifford multiplication. Many practical applications, however, possess additional algebraic structure in the input multivectors or the computational model. We analyze these distinct complexity regimes below.

Table 3: Computational regimes for QCM. Effective complexity includes the cost of postselection or amplitude amplification when required.

Regime	Assumption	Effective complexity
Dense inputs	Typical $p_0 = O(N^{-1})$	$O(N \log N)$
Amplitude amplification	$p_0 \ge 1/\mathrm{poly}(\log N)$	$O(\mathrm{poly}(\log N))$
Structured inputs	$k_A, k_B = \mathrm{poly}(\log N)$	$\mathrm{poly}(k_A, k_B, \log N)$

Amplitude amplification. Amplitude amplification assumes efficient implementations of reflections about both the prepared input state and the success subspace. Reflection about the success subspace is implemented by a phase flip conditioned on the ancilla register identifying the desired branch, while reflection about the prepared state is done by conjugating the standard reflection about $|0\rangle$ with the state-preparation circuit. Since the Clifford multiplication circuit has linear depth, each Grover iteration incurs only linear circuit overhead. Suppose a family of inputs has success probability $p_0 \ge \frac{1}{\mathrm{poly}(n)}$. Then the postselected circuit can be promoted to an efficient state-preparation procedure. Since the Clifford multiplication circuit has depth $O(n)$, naive repetition prepares the desired branch with expected depth $O\left(\frac{\log N}{p_0}\right)$. Amplitude amplification reduces the postselection overhead quadratically, giving expected depth

$O\left(\frac{n}{\sqrt{p_0}}\right) = O\left(\frac{\log N}{\sqrt{p_0}}\right)$, assuming efficient reflections about the prepared input state and the success subspace. Amplitude amplification makes polynomially small success probabilities efficient, but it does not convert exponentially small branch weight into a polynomial-depth algorithm.

Support-adapted projection. The factor $2^{-n/2}$ is because the algorithm projects the first register onto the uniform superposition

$$|+\rangle^{\otimes n} = 2^{-n/2} \sum_{x \in (\mathbb{Z}_2)^n} |x\rangle,$$

which averages uniformly over all 2^n blade indices. If only a small subset of basis blades has nonzero coefficients, projecting onto the full superposition averages over many irrelevant basis blades. Now suppose that $S_A = \mathrm{supp}(A)$ has cardinality s, and that the normalized support state

$$|S_A\rangle = s^{-1/2} \sum_{x \in S_A} |x\rangle$$

can be prepared efficiently. Projecting onto $|S_A\rangle$,

$$s^{-1/2} \sum_z \left(\sum_{x \in S_A} a_x b_{x \oplus z} \chi(x, x \oplus z) \right) |z\rangle = s^{-1/2} \sum_z c_z |z\rangle,$$

so the success probability becomes

$$p_S = s^{-1} \|AB\|_2^2.$$

When $s = \mathrm{poly}(n)$ and $\|AB\|_2^2 \geq 1/\mathrm{poly}(n)$, the postselection overhead is polynomial. Support-adapted projection replaces the ambient dimension 2^n by the effective support size s, but the improvement is conditional on the efficient preparation of the support state $|S_A\rangle$.

Sparse multivectors. If $A = \sum_{i=1}^{k_A} a_i e_{x_i}$, $\quad B = \sum_{j=1}^{k_B} b_j e_{y_j}$, where $k_A, k_B = \mathrm{poly}(n)$, then only $k_A k_B$ blade pairs contribute to the geometric product. Both the classical multiplication and the quantum state-preparation overhead are polynomial. Sparse multivectors provide a practically important setting in which the quantum multiplication primitive can be implemented without incurring exponential postselection overhead.

2.3 Application

Property estimation and decision problems.

The analysis identifies the cases in which quantum Clifford multiplication efficiently prepares the normalized product state

$$|AB\rangle = \frac{1}{\|AB\|_2} \sum_z c_z |z\rangle.$$

The most appropriate computational task is therefore not the reconstruction of every coefficient of the geometric product, but the efficient estimation of properties of the prepared state. As in the Harrow–Hassidim–Lloyd algorithm [14] and related quantum linear algebra algorithms, the product is accessed through efficiently measurable observables rather than complete classical output. This introduces a family of promise decision problems whose acceptance criterion depends on efficiently measurable properties of the Clifford product.

For example, letting $M_0 = |0^n\rangle\langle 0^n|$ denote projection onto the scalar blade,

$$\langle AB|M_0|AB\rangle = \frac{|\langle AB\rangle_0|^2}{\|AB\|_2^2},$$

one may verify whether the normalized scalar weight exceeds a prescribed threshold. Similarly, projection onto a fixed grade subspace determines whether the normalized grade-k weight exceeds a threshold, while more generally any efficiently implementable projector onto blades, grades, parity sectors, ideals, or other algebraically defined subspaces defines an associated promise decision problem.

Theorem 2.6 (Observable decision problems for QCM)**.** *Let $\{M_n\}$ be a uniform family of efficiently measurable observables. If the normalized Clifford product state $|AB\rangle$ is prepared with inverse-polynomial success probability, then every promise decision problem whose acceptance criterion is obtained by thresholding*

$$\langle AB|M_n|AB\rangle$$

with an inverse-polynomial promise gap belongs to BQP*. Every promise problem with a polynomial-time many-one or Turing reduction to estimating such expectation values belongs to* BQP*.*

Beyond observables defined directly by blades and grades, the same approach may be suited to decision problems from larger computational tasks whose solutions use Clifford algebras. Classical parameterized algorithms for Hamiltonian cycle, Steiner tree, feedback vertex set, and related graph problems on bounded-treewidth graphs employ repeated Clifford multiplication through noncommutative subset convolution. If the relevant intermediate quantities can be encoded as efficiently measurable observables of the prepared product state, the quantum Clifford multiplication algorithm could be used within quantum algorithms for these decision problems.

3 Outlook

The geometric product is a dense, noncommutative bilinear operation, yet its noncommutativity is encoded entirely by a cocycle over an underlying Abelian group. Knowing this, the computational difficulty of Clifford multiplication is not in the noncommutativity, but in the efficient implementation of its cocycle twist. This invites the question of which group algebras, twisted group algebras, crossed

products, and semisimple algebras have comparable quantum multiplication algorithms.

The work also points toward several practical directions. Since Clifford multiplication underpins rotors, spinors, versors, Clifford-valued differential operators, and geometric models of relativistic physics, an efficient multiplication primitive unlocks quantum algorithms devised natively in geometric algebra instead of through matrix representations. The close relationship between Clifford and matrix algebras also leaves open the extent to which these ideas extend to quantum matrix multiplication. Understanding these connections may uncover new quantum primitives for scientific computing, robotics, computer vision, spacetime simulation, and geometric numerical methods. Ultimately, the significance may lie not only in quantum Clifford multiplication itself, but in the wider computational direction it opens. The algorithm shows that harmonic methods need not be restricted to transforms; they may also provide efficient realizations of algebraic multiplication. Whether Clifford multiplication is an isolated example or the first member of a theory of harmonic quantum algebraic computation remains to be seen.

References

[1] R. Abłamowicz and B. Fauser. On computational complexity of clifford algebra. *Journal of Mathematical Physics*, 50(5):053514, 2009.

[2] H. Albuquerque and S. Majid. Clifford algebras obtained by twisting of group algebras. *Journal of Pure and Applied Algebra*, 171(2–3):133–148, 2002.

[3] Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 522–539. Society for Industrial and Applied Mathematics, 2021.

[4] Eduardo Bayro-Corrochano. *Geometric Computing for Perception Action Systems: Concepts, Algorithms and Scientific Applications*. Springer, 2001.

[5] Eduardo J. Bayro-Corrochano. Geometric neural computing. *IEEE Transactions on Neural Networks*, 12(5):968–986, 2001.

[6] Johann Brehmer, Pim de Haan, Víctor Garcia Satorras, and Taco Cohen. Geometric algebra transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

[7] Sven Buchholz and Gerald Sommer. On clifford neurons and clifford multilayer perceptrons. *Neural Networks*, 21(7):925–935, 2008.

[8] Grecia Castelazo, Quynh T. Nguyen, Giacomo De Palma, Dirk Englund, Seth Lloyd, and Bobak T. Kiani. Quantum algorithms for group convolution,

cross-correlation, and equivariant transformations. *Phys. Rev. A*, 106:032402, Sep 2022.

[9] William Kingdon Clifford. Applications of Grassmann's extensive algebra. *American Journal of Mathematics*, 1(4):350–358, 1878.

[10] Don Coppersmith. An approximate fourier transform useful in quantum factoring. Technical Report RC19642, IBM Research, 1994.

[11] Leo Dorst, Daniel Fontijne, and Stephen Mann. *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry*. Morgan Kaufmann, Burlington, MA, 2007.

[12] Hermann Grassmann. *Die Lineale Ausdehnungslehre, ein neuer Zweig der Mathematik*. Otto Wigand, Leipzig, 1844.

[13] Hermann Grassmann. *Die Ausdehnungslehre: Vollständig und in strenger Form bearbeitet*. Enslin, Berlin, 1862.

[14] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.

[15] David Hestenes. *Space-Time Algebra*. Gordon and Breach, New York, 2015.

[16] David Hestenes and Garret Sobczyk. *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Fundamental Theories of Physics. Springer, Dordrecht, 1984.

[17] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem. In *Electronic Colloquium on Computational Complexity (ECCC)*, 1995. Expanded version in arXiv:quant-ph/9511026.

[18] Chris Lomont. The hidden subgroup problem – review and open problems. *arXiv preprint quant-ph/0411037*, 2004.

[19] Pertti Lounesto. *Clifford Algebras and Spinors*. Cambridge University Press, 2 edition, 2001.

[20] Kagwe A. Muchane. Rotor-valued orientation as generalized clifford geometry, 2026. Manuscript in preparation.

[21] Kagwe A. Muchane. The state-operator clifford compatibility: A real algebraic framework for quantum information, 2026.

[22] David Ruhe, Johannes Brandstetter, and Patrick Forré. Clifford group equivariant neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 62922–62990. Curran Associates, Inc., 2023.

[23] David Ruhe, Jayesh K. Gupta, and Johannes Brandstetter. Geometric clifford algebra networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29461–29486. PMLR, 2023.

[24] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[25] Rich Wareham, Jonathan Cameron, and Joan Lasenby. Applications of conformal geometric algebra in computer vision and graphics. In *Computer Algebra and Geometric Algebra with Applications*, pages 329–349. Springer, 2005.