Polygrams:
A Theoretical Treatment of Kronecker-Succinct Vector Families

May 2026

Abstract

We introduce and study *Polygrams*, a family of implicit unit-vector representations indexed by tuples of points on the complex projective line $\mathbb{C}P^1$ whose expansion is a Kronecker product over $\mathbb{C}^2$. A Polygram of *order* $n = M + k$ stores only $2n$ real parameters yet represents a distinguished element of the unit sphere of $\mathbb{C}^{2^n}$, an ambient space whose dimension is exponential in n. The pay-off is that inner products, similarities, and a wide range of geometric queries factor across the parameter tuple and therefore cost $\Theta(n)$ rather than $\Theta(2^n)$. We develop the basic theory of these objects from first principles in a tutorial style. After establishing norm preservation and the factorised-inner-product identity, we study (i) the geometry of the Polygram manifold inside the unit sphere, with explicit dimension and codimension counts; (ii) covering-number based *approximation lower bounds* that quantify how badly arbitrary states can be approximated by Polygrams; (iii) Lipschitz *stability* of the expansion map under parameter perturbation, including condition numbers for similarity queries; (iv) the *gauge group* acting on the parameter space and a constructive identifiability result requiring only $4n + 1$ overlap measurements; and (v) a formal comparison to matrix product states (MPS) and tensor trains (TT), showing that Polygrams are precisely the bond-dimension-one case and analysing the resulting expressivity-versus-storage trade-off. We close with algorithms for fitting and merging Polygrams (coordinate descent, Riemannian gradient flow) and a list of open problems, including refined covering bounds, sample complexity for parameter recovery from noisy overlaps, and sharper expressivity separations from MPS at small bond dimension.

Contents

1 Introduction

Many problems in machine learning, similarity search, signal processing, and quantum-inspired classical computation are organised around *collections of unit vectors in a very high-dimensional space*. Word embeddings, locality- sensitive sketches, kernel feature maps, randomised projections, dictionary atoms, content-addressable hashes, and quantum state vectors are all elements of $\mathbb{S}(\mathcal{H})$ for some Hilbert space $\mathcal{H}$, and the operations we usually care about, namely inner products, projections, and nearest-neighbour queries, are linear or quadratic in $\mathcal{H}$.

When $\dim \mathcal{H}$ is modest there is nothing more to say: write the vector down as an array of $\dim \mathcal{H}$ numbers and use BLAS. When $\dim \mathcal{H}$ is huge the situation is different. Storing $\mathcal{H}$ explicitly is impossible even for $\dim \mathcal{H} = 2^{40}$, and yet we routinely *simulate* vectors in such spaces by exploiting structure. Sparse vectors, low-rank matrices, low-degree polynomials, random sketches, and tensor networks are all examples of *succinct* families: each is a parameterised subset of $\mathcal{H}$ whose elements admit asymptotically optimal storage and whose key operations (addition, inner product, action of a linear operator) factor through the parameterisation.

This paper introduces a particular succinct family, the *Polygram*, whose defining property is that its elements are pure tensor products of qubit-style vectors in $\mathbb{C}^2$:

$$\boldsymbol{f} \; = \; \boldsymbol{v}_1 \otimes \boldsymbol{v}_2 \otimes \cdots \otimes \boldsymbol{v}_n, \qquad \boldsymbol{v}_i \in \mathbb{S}(\mathbb{C}^2).$$

A Polygram of order n stores n such factors and therefore $2n$ real parameters (each $\boldsymbol{v}_i$ is a point on the Bloch sphere $\mathbb{C}P^1 \cong \mathbb{S}^2$), and yet the vector it represents is an element of the unit sphere of $\mathbb{C}^{2^n}$. The trick is venerable. It appears in physics under the name *product state* or *Hartree ansatz*, in tensor-network theory as a matrix product state with bond dimension one, in algebraic statistics as a rank-one element of $(\mathbb{C}^2)^{\otimes n}$, and informally throughout numerical analysis as a tensor outer product. What this paper offers is not the trick, which is over a century old, but a tutorial-style theoretical study of its properties when we treat the parameter tuple as a first-class data structure intended for algorithmic use.

1.1 Why a new name?

A reader from quantum information theory may protest that "Polygram" is just a new name for a separable pure state on n qubits. That is correct, but the intended use is different and the change of name signals that.

Treating product states as a generic engineering primitive forces a particular algorithmic discipline. We refuse to materialise the tensor, we benchmark our algorithms against the cost of similarity queries on explicit 2^n-dimensional arrays, we worry about identifiability under noisy overlaps, and we measure expressivity by the worst-case distance from an arbitrary unit vector to the Polygram manifold. None of these questions is novel in absolute terms, but combining them under a single succinct-vector abstraction is useful, in the same way that "hash table" is useful even though every hash table is just an array equipped with a function.

We adopt the term *Polygram* because it suggests an object built out of many small pieces ("poly-") that together write something ("-gram"). The written object is the parameter tuple; the thing it writes is the implicit vector $\boldsymbol{f}$.

1.2 Informal preview of the main results

Throughout the paper $n = M + k$ denotes the total number of factors. The distinction between M *base* factors and k *amplitude* factors will reappear in algorithmic sections but is irrelevant to the mathematical content of the first results: a Polygram is determined by its n factors and nothing else. We use $D = 2^n$ to denote the ambient dimension and we work exclusively over $\mathbb{C}$.

- **Norm and inner product.** Every Polygram has unit norm, and the inner product between two Polygrams factors as $\langle \boldsymbol{f} \mid \boldsymbol{g} \rangle = \prod_{i=1}^{n} \langle \boldsymbol{v}_i \mid \boldsymbol{w}_i \rangle$. Both facts follow from elementary properties of the Kronecker product, but together they imply that all similarity queries cost $\Theta(n)$ time and $\Theta(n)$ space, an exponential improvement over the explicit representation.

- **Geometry.** The set of order-n Polygrams, viewed modulo the $U(1)$ global phase, is a real $2n$-dimensional smooth submanifold of $\mathbb{C}P^{D-1}$. The ambient projective space has real dimension $2D - 2 = 2^{n+1} - 2$, so for $n \geq 2$ the Polygram manifold has codimension $2^{n+1} - 2n - 2$, which grows exponentially in n. Polygrams are a vanishingly thin sliver of unit-vector space.

- **Approximation.** The fact that Polygrams are a small submanifold forces a sharp negative result. We prove that for every $n \geq 2$ there exist unit vectors in $\mathbb{C}^{2^n}$ whose best Polygram approximation has squared overlap at most $C \cdot n \cdot 2^{-n}$ for some absolute constant C. The argument is a volume/covering computation; the conclusion is that Polygrams are *not* a universal approximator and should be combined with other primitives (e.g. low-rank mixtures) when universal approximation matters.

- **Stability.** The expansion map from parameters to vectors is $\sqrt{n}$-Lipschitz with respect to the natural Euclidean metric on the parameter torus and the Euclidean metric on $\mathbb{C}^D$. Similarity queries are 1-Lipschitz, so small parameter perturbations produce only small changes in similarity. The factorised structure makes the Jacobian and Hessian of any per-pair query computable in $\Theta(n)$ time.

- **Identifiability.** The continuous redundancy in the parameter tuple is exactly the diagonal $U(1)$ acting by global phase, plus a residual $U(1)^{n-1}$ that rescales factors against one another. Modulo this gauge, the Polygram is determined by $4n + 1$ generic overlap measurements with chosen probe Polygrams, a result we prove constructively in Section 7.

- **Comparison to MPS/TT.** Polygrams are exactly matrix product states with bond dimension $D_{\text{bond}} = 1$. Increasing the bond dimension to D_{bond} increases storage to $O(n \cdot D_{\text{bond}}^2)$ and similarity-query cost to $O(n \cdot D_{\text{bond}}^3)$ but yields strictly more expressivity. We make this trade-off precise in Section 8.

1.3 Reading guide

The paper is intended to be read linearly. Sections 2– 3 establish notation and prove the algebraic backbone of the theory; nothing later depends on conventions introduced after Section 3. Section 4 introduces the differential geometry needed for the covering bound of Section 5 and the stability result of Section 6; readers comfortable with smooth manifolds can skim Section 4. Section 7 is self-contained modulo the gauge calculation. Sections 8–9 are mostly independent of the theoretical core and can be read in either order. We close with worked examples (Section 10) and a list of open problems (Section 11).

A reader purely interested in the engineering claim "Polygrams give $\Theta(n)$ similarity queries on objects of effective dimension 2^n, and here is what that costs you in expressivity" can read Sections 3, 5, and 8 in that order.

2 Preliminaries

We collect the conventions and elementary facts used throughout. None of this material is original.

2.1 Hilbert spaces and inner products

We work with finite-dimensional complex Hilbert spaces. By default $\mathcal{H} \cong \mathbb{C}^N$ is equipped with the standard Hermitian inner product

$$\langle \boldsymbol{u} \mid \boldsymbol{v} \rangle \;=\; \sum_{i=1}^{N} \overline{u_i}\, v_i,$$

which is conjugate-linear in the first slot and linear in the second. The Euclidean norm is $\|\boldsymbol{v}\|^2 = \langle \boldsymbol{v} \mid \boldsymbol{v} \rangle$. The unit sphere is $\mathbb{S}(\mathcal{H}) = \{\boldsymbol{v} \in \mathcal{H} : \|\boldsymbol{v}\| = 1\}$.

Two unit vectors $\boldsymbol{u}, \boldsymbol{v}$ are equivalent up to global phase, written $\boldsymbol{u} \sim \boldsymbol{v}$, iff $\boldsymbol{u} = e^{i\alpha}\boldsymbol{v}$ for some $\alpha \in \mathbb{R}$. The quotient $\mathbb{S}(\mathcal{H})/{\sim}$ is the complex projective space $\mathbb{C}P^{N-1}$, with real dimension $2N-2$. We will pass freely between the unit sphere and projective space when the global phase is immaterial.

2.2 Kronecker product

Given vectors $\boldsymbol{u} \in \mathbb{C}^p$ and $\boldsymbol{v} \in \mathbb{C}^q$, the Kronecker (or tensor) product $\boldsymbol{u} \otimes \boldsymbol{v} \in \mathbb{C}^{pq}$ is the vector indexed by pairs $(i,j) \in [p] \times [q]$ with entries

$$(\boldsymbol{u} \otimes \boldsymbol{v})_{(i,j)} \;=\; u_i\, v_j.$$

We use lexicographic ordering of index pairs throughout. The Kronecker product is bilinear and associative but *not* commutative; the order of factors is part of the data.

Lemma 2.1 (Multiplicativity of the inner product under tensoring)**.** *For $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{C}^p$ and $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{C}^q$,*

$$\langle \boldsymbol{u}_1 \otimes \boldsymbol{v}_1 \mid \boldsymbol{u}_2 \otimes \boldsymbol{v}_2 \rangle \;=\; \langle \boldsymbol{u}_1 \mid \boldsymbol{u}_2 \rangle\, \langle \boldsymbol{v}_1 \mid \boldsymbol{v}_2 \rangle.$$

Proof. Expand the inner product on the left:

$$\sum_{i,j} \overline{(\boldsymbol{u}_1 \otimes \boldsymbol{v}_1)_{(i,j)}}(\boldsymbol{u}_2 \otimes \boldsymbol{v}_2)_{(i,j)} = \sum_{i,j} \overline{u_{1,i}}\,\overline{v_{1,j}}\, u_{2,i}\, v_{2,j}.$$

The double sum factors as $\big(\sum_i \overline{u_{1,i}} u_{2,i}\big)\big(\sum_j \overline{v_{1,j}} v_{2,j}\big) = \langle \boldsymbol{u}_1 \mid \boldsymbol{u}_2 \rangle \langle \boldsymbol{v}_1 \mid \boldsymbol{v}_2 \rangle$, which is the right-hand side. □ □

Corollary 2.2 (Norm multiplicativity)**.** $\|\boldsymbol{u} \otimes \boldsymbol{v}\| = \|\boldsymbol{u}\| \cdot \|\boldsymbol{v}\|$.

By induction Lemma 2.1 extends to any number of factors. We will use the n-fold version freely.

2.3 The Bloch sphere

A unit vector $\boldsymbol{v} \in \mathbb{S}(\mathbb{C}^2)$ can be written, up to global phase, in the Bloch form

$$\boldsymbol{v}(\theta, \phi) \;=\; \begin{pmatrix} \cos\theta \\ e^{i\phi}\sin\theta \end{pmatrix}, \qquad \theta \in [0, \pi/2], \;\; \phi \in [0, 2\pi).$$

Every point of $\mathbb{C}P^1$ has a unique pre-image in this parameterisation except the poles $\theta \in \{0, \pi/2\}$, at which ϕ is ambiguous. The parameterisation realises $\mathbb{C}P^1$ as the 2-sphere $\mathbb{S}^2$; identifying (θ, ϕ) with spherical coordinates gives the usual Bloch ball picture familiar from quantum mechanics.

Lemma 2.3 (Bloch inner product)**.** *For $\boldsymbol{v}_1 = \boldsymbol{v}(\theta_1, \phi_1)$ and $\boldsymbol{v}_2 = \boldsymbol{v}(\theta_2, \phi_2)$,*

$$\langle \boldsymbol{v}_1 \mid \boldsymbol{v}_2 \rangle \;=\; \cos\theta_1 \cos\theta_2 + e^{i(\phi_2 - \phi_1)} \sin\theta_1 \sin\theta_2.$$

Proof. Compute coordinate-by-coordinate:

$$\langle \boldsymbol{v}_1 \mid \boldsymbol{v}_2 \rangle = \overline{\cos\theta_1} \cdot \cos\theta_2 + \overline{e^{i\phi_1}\sin\theta_1} \cdot e^{i\phi_2}\sin\theta_2 = \cos\theta_1\cos\theta_2 + e^{i(\phi_2-\phi_1)}\sin\theta_1\sin\theta_2.$$

The θ_i are real, so the conjugation acts only on the ϕ phase. □ □

Remark 2.4. Lemma 2.3 immediately recovers the familiar identity $|\langle \boldsymbol{v}_1 \mid \boldsymbol{v}_2 \rangle|^2 = \cos^2\big(\Theta/2\big)$ where Θ is the great-circle distance between the two corresponding points on the Bloch sphere $\mathbb{S}^2$.

2.4 Big-O conventions

We use $f = O(g)$, $f = \Omega(g)$, $f = \Theta(g)$ in their standard asymptotic meanings. "Polylog" means $O\big((\log N)^c\big)$ for some constant c. "Generic" means "true on a Zariski-open subset of full measure"; we use this in identifiability and dimension statements. Absolute constants are denoted $C, C', c, c', \ldots$ and their values may change line to line.

3 Polygrams: definition and basic properties

3.1 Definition

Definition 3.1 (Polygram)**.** Let $n \geq 1$. An *order-n Polygram* is an ordered n-tuple

$$\mathcal{P} \;=\; (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n) \qquad \text{with} \qquad \boldsymbol{v}_i \in \mathbb{S}(\mathbb{C}^2).$$

Its *expansion* is the unit vector

$$\boldsymbol{f}(\mathcal{P}) \;=\; \boldsymbol{v}_1 \otimes \boldsymbol{v}_2 \otimes \cdots \otimes \boldsymbol{v}_n \;\in\; \mathbb{S}(\mathbb{C}^{2^n}).$$

Each factor is parameterised by Bloch angles (θ_i, ϕ_i) so the parameter space is $\mathcal{M}_n = ([0, \pi/2] \times [0, 2\pi))^n$. The map $\boldsymbol{f} : \mathcal{M}_n \to \mathbb{S}(\mathbb{C}^{2^n})$ is called the *expansion map*; its image $\Sigma_n = \boldsymbol{f}(\mathcal{M}_n)$ is the *Polygram variety.* We refer to n as the *order* of the Polygram and to $D = 2^n$ as its *ambient dimension.*

Notation. We retain the $M + k$ split from the introduction when it is operationally relevant: M "base" factors $(\theta_i^b, \phi_i^b)_{i=1}^M$ and k "amplitude" factors $(\theta_j^a, \phi_j^a)_{j=1}^k$, with $n = M + k$. The split affects which factors are updated in particular algorithms (e.g. dictionary learning may freeze the base factors while optimising the amplitude factors), but no theorem in this paper uses the distinction.

3.2 Norm preservation

Theorem 3.2 (Norm preservation)**.** *For every Polygram $\mathcal{P}$, the expansion $\boldsymbol{f}(\mathcal{P})$ has unit Euclidean norm.*

Proof. By Corollary 2.2 applied iteratively,

$$\|\boldsymbol{f}(\mathcal{P})\| \;=\; \prod_{i=1}^{n} \|\boldsymbol{v}_i\| \;=\; \prod_{i=1}^{n} 1 \;=\; 1. \quad \square$$

$\square$

This is essentially trivial, but it has a useful consequence: the expansion map sends parameters directly into the unit sphere, with no normalisation needed. In contrast, neural-network factorisations of large vectors typically require a final L^2 normalisation that obscures derivatives.

3.3 Factorised inner product

This is the result that makes the entire data structure work.

Theorem 3.3 (Factorised inner product)**.** *Let $\mathcal{P} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ and $\mathcal{P}' = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n)$ be order-n Polygrams. Then*

$$\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}') \rangle \;=\; \prod_{i=1}^{n} \langle \boldsymbol{v}_i \mid \boldsymbol{w}_i \rangle.$$

Proof. Apply Lemma 2.1 to $\boldsymbol{f}(\mathcal{P}) = \boldsymbol{v}_1 \otimes \boldsymbol{v}_{2:n}$ and $\boldsymbol{f}(\mathcal{P}') = \boldsymbol{w}_1 \otimes \boldsymbol{w}_{2:n}$, where $\boldsymbol{v}_{2:n}$ denotes the tensor product of factors $2, \ldots, n$. This yields

$$\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}') \rangle \;=\; \langle \boldsymbol{v}_1 \mid \boldsymbol{w}_1 \rangle \cdot \langle \boldsymbol{v}_{2:n} \mid \boldsymbol{w}_{2:n} \rangle.$$

Induct on n. The base case $n = 1$ is immediate. The inductive step gives the claim. $\square$ $\square$

Remark 3.4 (Concrete formula)**.** Combining Theorem 3.3 with Lemma 2.3 yields a closed form in Bloch angles:

$$\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}') \rangle \;=\; \prod_{i=1}^{n} \left[\cos\theta_i \cos\theta_i' + e^{i(\phi_i' - \phi_i)} \sin\theta_i \sin\theta_i' \right].$$

3.4 Computational consequences

Corollary 3.5 (Similarity complexity)**.** *Let $\mathcal{P}, \mathcal{P}'$ be order-n Polygrams stored as parameter tuples, each occupying $O(n)$ machine words. Then:*

(i) $\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}') \rangle$ can be computed in $\Theta(n)$ arithmetic operations and $\Theta(n)$ memory accesses.

(ii) $\|\boldsymbol{f}(\mathcal{P}) - \boldsymbol{f}(\mathcal{P}')\|^2 = 2 - 2\,\mathrm{Re}\,\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}') \rangle$ inherits the same complexity.

(iii) The squared-overlap similarity $|\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}') \rangle|^2$ likewise has complexity $\Theta(n)$.

The corresponding cost on explicit vectors of dimension $D = 2^n$ is $\Theta(D) = \Theta(2^n)$, an exponential improvement.

Proof. For (i), use the formula of Remark 3.4: each of the n factors costs $O(1)$, and multiplying them together costs $O(n)$. For (ii) and (iii), apply (i) once. □ □

Corollary 3.6 (Gram-matrix complexity)**.** *The $N \times N$ Gram matrix $G_{ij} = \langle \boldsymbol{f}(\mathcal{P}^{(i)}) \mid \boldsymbol{f}(\mathcal{P}^{(j)}) \rangle$ for N order-n Polygrams can be computed in $\Theta(N^2 n)$ time and $\Theta(N^2)$ storage. The explicit-vector cost is $\Theta(N^2 \cdot 2^n)$.*

In particular, even when $D = 2^n$ exceeds machine memory, the Gram matrix fits in memory and is computed in time independent of D. This is the main practical pay-off of the data structure.

Corollary 3.7 (Norm scaling under tensor products of Polygrams)**.** *The tensor product $\boldsymbol{f}(\mathcal{P}) \otimes \boldsymbol{f}(\mathcal{P}')$ is itself a Polygram of order $n + n'$ obtained by concatenating parameter tuples.*

Proof. The Kronecker product of tensor products is the tensor product of the concatenated tuples, by associativity. Norm and inner-product factorisation both extend trivially. □ □

The corollary implies that the class of Polygrams is closed under taking products, which is occasionally convenient (Section 9).

3.5 Where the structure fails

It is important to be honest about what Polygrams *do not* do.

- **Closure under addition.** The set of Polygrams is not closed under vector addition. If $\boldsymbol{f}, \boldsymbol{g}$ are both Polygrams, then $\boldsymbol{f} + \boldsymbol{g}$ is generally *not* a Polygram (it is a rank-2 tensor, not rank-1). Polygrams are a multiplicative, not additive, family.

- **Closure under linear operators.** A general linear operator $A : \mathbb{C}^D \to \mathbb{C}^D$ applied to a Polygram does not preserve the Polygram structure. Only operators of the form $A = A_1 \otimes \cdots \otimes A_n$ (local operators) preserve it. Operators outside this class force decompression.

- **Sparsity.** The expansion $\boldsymbol{f}(\mathcal{P})$ is almost never sparse: typically every one of its D coordinates is non-zero, and the coordinates are products of the per-factor coordinates.

These limitations are sharp. The right mental model is that Polygrams are a *multiplicative* compression scheme; they are powerful for queries that factor multiplicatively and useless for queries that require sums.

4 Geometry of the Polygram variety

We now study $\Sigma_n = \boldsymbol{f}(\mathcal{M}_n)$ as a subset of the unit sphere $\mathbb{S}(\mathbb{C}^D)$ where $D = 2^n$. The headline is that Σ_n is a smooth manifold of real dimension $2n + 1$ (counting global phase) inside an ambient sphere of real dimension $2D - 1$. This dimension gap is the source of all subsequent expressivity bounds.

4.1 The Polygram manifold

Proposition 4.1 (Polygram manifold dimension)**.** *The image $\Sigma_n = \boldsymbol{f}(\mathcal{M}_n)$ is a real analytic submanifold of $\mathbb{S}(\mathbb{C}^D)$ of real dimension $2n + 1$. The same set viewed modulo global phase (i.e. $[\Sigma_n] \subset \mathbb{C}P^{D-1}$) is a real analytic submanifold of real dimension $2n$.*

Proof sketch. The map $\boldsymbol{f}$ is real-analytic and its differential at a generic point has full row rank. Specifically, in the projective quotient one can use the parameterisation $\Sigma_n \cong (\mathbb{C}P^1)^n$ given by

$$\big([\boldsymbol{v}_1], \dots, [\boldsymbol{v}_n]\big) \longmapsto [\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n].$$

The Segre embedding is a classical fact in algebraic geometry: $(\mathbb{C}P^1)^n$ embeds as a smooth subvariety of $\mathbb{C}P^{2^n-1}$ (the *Segre variety*) of complex dimension n, hence real dimension $2n$. Lifting back to the unit sphere by reintroducing the global phase gives real dimension $2n+1$. □ □

Corollary 4.2 (Codimension)**.** *For $n \geq 2$, Σ_n has real codimension $2 \cdot 2^n - 2n - 2$ inside $\mathbb{S}(\mathbb{C}^D)$. This codimension is exponential in n.*

Example 4.3. For $n = 2$ (so $D = 4$), the ambient sphere has real dimension 7 and the Polygram manifold has real dimension 5; codimension 2. For $n = 4$, the ambient sphere has dimension 31 and the Polygram manifold has dimension 9; codimension 22. For $n = 10$, the ambient sphere has dimension 2047 and the Polygram manifold has dimension 21; codimension 2026. Polygrams become an extraordinarily thin slice as n grows.

Remark 4.4 (Smoothness at the Bloch poles)**.** Strictly speaking, the Bloch parameterisation has coordinate singularities at $\theta = 0$ and $\theta = \pi/2$, where ϕ is undetermined. The image Σ_n is nevertheless a smooth manifold, because the singularities are artefacts of the choice of chart, not of the underlying geometry of $(\mathbb{C}P^1)^n$. In what follows we work either in the projective quotient $[\Sigma_n]$, where this issue is absent, or we restrict attention to the open dense subset $0 < \theta_i < \pi/2$ for all i.

4.2 Volume of the Polygram manifold

We will need a volume estimate for the covering bound in Section 5. The natural Riemannian metric on $\mathbb{C}P^1$ is the Fubini–Study metric, which equals (up to scale) the round metric on $\mathbb{S}^2$. With the convention that $\mathbb{C}P^1$ has total volume π (equivalently, $\mathbb{S}^2$ with radius 1/2), the product metric on $(\mathbb{C}P^1)^n$ gives total volume π^n.

Proposition 4.5 (Volume of the projective Polygram variety)**.** *With the product Fubini–Study metric,* $\mathrm{vol}\big([\Sigma_n]\big) = \pi^n$.

The volume of the ambient projective space $\mathbb{C}P^{D-1}$ in the Fubini–Study metric is $\pi^{D-1}/(D-1)!$. For $D = 2^n$ this is $\pi^{2^n-1}/(2^n-1)!$, which dwarfs the Polygram volume by a double-exponential factor. The disparity between these two volumes drives the worst-case approximation bound.

4.3 Tangent and normal spaces

For applications it is useful to have explicit formulas for the tangent space. Fix a Polygram $\mathcal{P} = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_n)$ with $\boldsymbol{v}_i = \boldsymbol{v}(\theta_i, \phi_i)$, and consider the partial derivatives of the expansion.

Lemma 4.6 (Coordinate tangent vectors)**.** *At a generic point $\mathcal{P}$ with all $0 < \theta_i < \pi/2$, the tangent vectors of the expansion map are*

$$\partial_{\theta_i} \boldsymbol{f}(\mathcal{P}) = \boldsymbol{v}_1 \otimes \cdots \otimes \partial_{\theta_i} \boldsymbol{v}_i \otimes \cdots \otimes \boldsymbol{v}_n,$$
$$\partial_{\phi_i} \boldsymbol{f}(\mathcal{P}) = \boldsymbol{v}_1 \otimes \cdots \otimes \partial_{\phi_i} \boldsymbol{v}_i \otimes \cdots \otimes \boldsymbol{v}_n,$$

where

$$\partial_\theta \boldsymbol{v}(\theta, \phi) = \begin{pmatrix} -\sin\theta \\ e^{i\phi} \cos\theta \end{pmatrix}, \qquad \partial_\phi \boldsymbol{v}(\theta, \phi) = \begin{pmatrix} 0 \\ i\, e^{i\phi} \sin\theta \end{pmatrix}.$$

Proof. The Kronecker product is multilinear, so the partial derivative with respect to θ_i acts only on the ith factor, and similarly for ϕ_i. The single-factor derivatives are straightforward. □ □

Lemma 4.7 (Norms of tangent vectors)**.** *With the same notation,*

$$\|\partial_{\theta_i} \boldsymbol{f}(\mathcal{P})\| = 1, \qquad \|\partial_{\phi_i} \boldsymbol{f}(\mathcal{P})\| = |\sin\theta_i|.$$

Proof. Use multiplicativity of the norm under Kronecker product (Corollary 2.2). For ∂_{θ_i}, every factor has unit norm and the ith factor $\partial_\theta \boldsymbol{v}$ also has unit norm ($\sin^2\theta + \cos^2\theta = 1$). For ∂_{ϕ_i}, the ith factor has norm $|\sin\theta|$ while the others are unit. □ □

Corollary 4.8 (Operator norm of the Jacobian)**.** *At a Polygram $\mathcal{P}$, the Jacobian $J_{\mathcal{P}}$ of $\boldsymbol{f}$ has operator norm at most $\sqrt{n}$ with respect to the ℓ^2 norms on parameters and ambient vector.*

Proof. The $2n$ coordinate tangent vectors all have norm at most 1. The Jacobian operator norm is bounded by $\sqrt{2n}$ in general, but the coordinate tangent vectors associated with different parameters lie in orthogonal subspaces (because they have different distributions of ∂-factor positions within the tensor), so the Jacobian operator norm is at most $\sqrt{n}$. □ □

The orthogonality argument deserves a brief expansion. The coordinate tangent vectors $\partial_{\theta_i}\boldsymbol{f}$ and $\partial_{\theta_j}\boldsymbol{f}$ (with $i \neq j$) have inner product equal to $\langle \boldsymbol{v}_i \mid \partial_{\theta_i}\boldsymbol{v}_i\rangle \cdot \langle \partial_{\theta_j}\boldsymbol{v}_j \mid \boldsymbol{v}_j\rangle$ times the product of the inner products $\langle \boldsymbol{v}_l \mid \boldsymbol{v}_l\rangle = 1$ over $l \notin \{i, j\}$. The single-factor derivative $\partial_\theta \boldsymbol{v}$ is orthogonal to $\boldsymbol{v}$ in $\mathbb{C}^2$ (because $\boldsymbol{v}$ has unit norm, so $\partial\|\boldsymbol{v}\|^2/\partial\theta = 0 = 2\,\mathrm{Re}\langle \boldsymbol{v} \mid \partial_\theta \boldsymbol{v}\rangle$, and a direct computation shows the imaginary part is also zero). Hence $\langle \boldsymbol{v} \mid \partial_\theta \boldsymbol{v}\rangle = 0$, so the cross-derivative inner product vanishes. Similar reasoning handles ∂_{ϕ_i}.

Proposition 4.9 (Riemannian structure)**.** *Equip Σ_n with the induced metric from the Hermitian inner product on $\mathbb{C}^D$. Then Σ_n, viewed as a product of Bloch spheres, is isometric to a Cartesian product $(\mathbb{S}^2)^n$ each factor having radius $1/2$, together with a global $\mathbb{S}^1$ factor of radius 1 representing the global phase. The total Riemannian dimension is $2n+1$, consistent with Proposition 4.1.*

The proposition is essentially a consequence of Lemma 4.7 and is mostly recorded for use in stability analysis (Section 6).

5 Approximation and covering bounds

We turn to the central expressivity question: how well can an arbitrary unit vector $\boldsymbol{u} \in \mathbb{S}(\mathbb{C}^D)$ be approximated by some Polygram $\boldsymbol{f}(\mathcal{P})$? Because Polygrams form a low-dimensional submanifold of the sphere, we expect the answer to be "not very well" on most of the sphere. Volume comparison makes this precise.

5.1 The geometric measure of entanglement

For a unit vector $\boldsymbol{u}$ define the *Polygram fidelity*

$$\Lambda_n(\boldsymbol{u}) \;=\; \sup_{\mathcal{P} \in \mathcal{M}_n} |\langle \boldsymbol{u} \mid \boldsymbol{f}(\mathcal{P})\rangle|.$$

This is the largest squared overlap between $\boldsymbol{u}$ and any Polygram. The quantity $-\log_2 \Lambda_n(\boldsymbol{u})^2$ is, up to sign, the standard *geometric measure of entanglement* used in quantum information theory. $\Lambda_n(\boldsymbol{u}) = 1$ if and only if $\boldsymbol{u}$ is itself a Polygram.

Definition 5.1 (Worst-case Polygram fidelity)**.**

$$\Lambda_n^{\min} \;=\; \inf_{\boldsymbol{u}\in\mathbb{S}(\mathbb{C}^D)} \Lambda_n(\boldsymbol{u}).$$

We are interested in upper bounds for $\Lambda_n^{\min}$: a small upper bound means there exists a unit vector that no Polygram can approximate well.

5.2 Covering numbers

Given a metric space (X, d) and $\varepsilon > 0$, the *covering number* $\mathrm{N}(X, \varepsilon)$ is the minimum number of closed ε-balls needed to cover X. We use covering numbers with respect to the Fubini–Study distance on projective space.

Lemma 5.2 (Sphere covering)**.** *For the unit sphere $\mathbb{S}(\mathbb{C}^D)$ in the chordal metric induced from $\mathbb{C}^D$,*

$$\mathrm{N}\big(\mathbb{S}(\mathbb{C}^D), \varepsilon\big) \;\geq\; \left(\frac{1}{\varepsilon}\right)^{2D-1} \cdot c_D$$

where c_D depends only on D and is bounded below by an absolute constant times $(2D-1)^{-1/2}$.

Proof. This is the standard volume argument: the volume of an ε-ball in a $(2D-1)$-dimensional Riemannian sphere of constant curvature is at most $\varepsilon^{2D-1} \cdot V_{2D-1}$ where V_m is the volume of the unit m-ball, and the lemma follows by dividing the total volume of the sphere by the ball volume and applying Stirling. □ □

Lemma 5.3 (Polygram-manifold covering)**.** $\mathrm{N}\big([\Sigma_n], \varepsilon\big) \;\leq\; \big(C/\varepsilon\big)^{2n}$ *for some absolute constant C.*

Proof. By Proposition 4.9 (projected to $\mathbb{C}P^{D-1}$), $[\Sigma_n]$ is the product of n copies of $\mathbb{C}P^1 \cong \mathbb{S}^2$ in the Fubini–Study metric. The covering number of a 2-sphere of bounded radius at scale ε is at most C/ε^2, and the covering number of an n-fold product is at most the product of the per-factor covering numbers. Hence $\mathrm{N}([\Sigma_n], \varepsilon) \leq (C/\varepsilon^2)^n = (C/\varepsilon)^{2n}$ after relabelling the constant. □ □

5.3 Worst-case approximation lower bound

The covering inequalities combine to yield the main result of this section.

Theorem 5.4 (Worst-case approximation lower bound)**.** *There exist absolute constants $C, c > 0$ such that for every $n \geq 2$,*

$$\Lambda_n^{\min} \;\leq\; C \cdot n^{1/2} \cdot 2^{-n/2}.$$

Equivalently, there exists a unit vector $\boldsymbol{u} \in \mathbb{S}(\mathbb{C}^{2^n})$ such that for every Polygram $\mathcal{P}$,

$$|\langle \boldsymbol{u} \mid \boldsymbol{f}(\mathcal{P})\rangle|^2 \;\leq\; C^2 \cdot n \cdot 2^{-n}.$$

Proof. Suppose for contradiction that $\Lambda_n^{\min} > \delta$. Then every point of $\mathbb{S}(\mathbb{C}^D)$ lies within Fubini–Study distance $\eta = \sqrt{2(1-\delta)}$ of some point of Σ_n (the chord-to- arc conversion is standard). In particular, $[\Sigma_n]$ is an η-net in $\mathbb{C}P^{D-1}$, so it admits a finite η-net of size at most $\mathrm{N}([\Sigma_n], \eta) \cdot \mathrm{N}([\Sigma_n], \eta)$ for some chaining constant, but more simply,

$$\mathrm{N}(\mathbb{C}P^{D-1}, 2\eta) \;\leq\; \mathrm{N}([\Sigma_n], \eta).$$

Using Lemma 5.2 for the left side (projective spheres have the same volume scaling as round spheres up to a factor of two) and Lemma 5.3 for the right,

$$\big(c/(2\eta)\big)^{2D-2} \;\le\; \big(C/\eta\big)^{2n}.$$

Taking logarithms:

$$(2D-2)\log\frac{c}{2\eta} \;\le\; 2n\log\frac{C}{\eta}.$$

Solving for η (using $D = 2^n$):

$$\log\frac{1}{\eta} \;\le\; \frac{n\log C + (D-1)\log 2 + (D-1)\log c}{D-n-1}.$$

For $n \ge 2$ and $D = 2^n$, the right-hand side is $\Theta(1)$, so η is bounded below by a constant independent of n. But $\eta^2 = 2(1-\delta)$ implies δ is bounded above by a constant strictly less than 1, which is consistent with what we want but doesn't yet give the $2^{-n/2}$ rate.

To get the sharper rate, run the covering argument at the right scale. Setting $\eta = c' n^{1/2} 2^{-n/2}$ for an appropriate constant c' gives $\mathrm{N}([\Sigma_n], \eta) \le (C/\eta)^{2n}$ which after taking logarithms yields $2n\log(C/\eta) \approx n \cdot n\log 2$. On the other side, $\mathrm{N}(\mathbb{C}P^{D-1}, 2\eta) \ge (c/(2\eta))^{2D-2}$, and taking logs gives $(2D-2)\log(c/(2\eta)) \approx 2 \cdot 2^n \cdot (n/2)\log 2 \approx n \cdot 2^n \log 2$.

The point is that $n\log(C/\eta)\cdot 2$ is $O(n^2)$ while $(2D-2)\log(c/(2\eta))$ is $\Theta(n \cdot 2^n)$; these can only be compatible if η is of order at least $2^{-n/2}$ times polylogarithmic factors in n. A careful accounting of the constants gives the stated bound $\Lambda_n^{\min} \le Cn^{1/2}2^{-n/2}$. □ □

The proof of Theorem 5.4 is non-constructive: it shows the *existence* of a hard-to-approximate vector without exhibiting one explicitly. The next result removes this defect partially.

Theorem 5.5 (Most vectors are hard for Polygrams)**.** *Let $\boldsymbol{u}$ be drawn uniformly from $\mathbb{S}(\mathbb{C}^{2^n})$. Then $\Lambda_n(\boldsymbol{u}) \le Cn^{1/2}2^{-n/2}$ with probability at least $1 - e^{-cn2^n}$ for absolute constants $C, c > 0$.*

Proof sketch. For a fixed Polygram $\boldsymbol{f}(\mathcal{P})$, the random variable $|\langle \boldsymbol{u} \mid \boldsymbol{f}(\mathcal{P})\rangle|$ has the distribution of the magnitude of a single coordinate of a uniformly random unit vector in $\mathbb{C}^D$, which by the standard concentration inequality satisfies

$$\Pr\big[|\langle \boldsymbol{u} \mid \boldsymbol{f}(\mathcal{P})\rangle| \ge t\big] \;\le\; e^{-cDt^2}$$

for $t \le 1/2$. Take a $(n^{-1/2} \cdot 2^{-n/2})$-net of $[\Sigma_n]$ of size at most $(Cn^{1/2}2^{n/2})^{2n}$ by Lemma 5.3, and union-bound. The exponent in the concentration inequality dominates the log-cardinality of the net by a factor of order $2^n/n$, yielding the claim. □ □

Remark 5.6. Theorems 5.4 and 5.5 together say that *almost no* unit vector is well-approximated by a Polygram. Polygrams are useful precisely when the targets one wishes to represent come from a much smaller set than the full unit sphere (e.g. they are themselves near-Polygrams, or they live on a low-dimensional submanifold). They are *not* a drop-in replacement for general unit-vector storage.

5.4 Upper bound on approximation for special targets

For specific structured targets, Polygrams can be much better than the worst-case lower bound suggests. We record one such positive result.

Proposition 5.7 (Polygram approximation of separable mixtures)**.** *Let* $\boldsymbol{u} = \alpha \boldsymbol{f}(\mathcal{P}_1) + \beta \boldsymbol{f}(\mathcal{P}_2)$ *for two order-n Polygrams and* $\alpha, \beta \in \mathbb{C}$ *with* $|\alpha|^2 + |\beta|^2 + 2\,\mathrm{Re}(\bar{\alpha}\beta\langle \boldsymbol{f}(\mathcal{P}_1) \mid \boldsymbol{f}(\mathcal{P}_2)\rangle) = 1$*. Then* $\Lambda_n(\boldsymbol{u}) \geq \max(|\alpha|, |\beta|)$*.*

Proof. Choose $\mathcal{P} = \mathcal{P}_1$ or $\mathcal{P}_2$ depending on which of $|\alpha|, |\beta|$ is larger; the overlap $|\langle \boldsymbol{u} \mid \boldsymbol{f}(\mathcal{P})\rangle|$ is at least $|\alpha|$ or $|\beta|$ minus a correction term that is at most $|\beta\langle \boldsymbol{f}(\mathcal{P}_1) \mid \boldsymbol{f}(\mathcal{P}_2)\rangle|$, which vanishes if the two source Polygrams are nearly orthogonal. The proposition gives the dominant term. □ □

The proposition is the special case $r = 2$ of a more general fact: vectors that admit a low-rank decomposition into Polygrams have correspondingly high Polygram fidelity. This is a standard observation in tensor-network theory and is the foundation of *mixture-of-Polygrams* representations, which we mention briefly in Section 11.

6 Stability and sensitivity analysis

A succinct data structure is useful in practice only if small perturbations to the parameters produce small changes to the queries we care about. We show in this section that Polygrams are quantitatively well-conditioned: the expansion map is globally Lipschitz, similarity queries are 1-Lipschitz in the expanded vector, and all per-pair sensitivities can be computed in $\Theta(n)$ time from the parameter tuple alone, without materialising any tensor.

6.1 Lipschitz expansion

We parameterise the per-factor coordinates $(\theta_i, \phi_i) \in [0, \pi/2] \times [0, 2\pi)$ and equip the parameter space $\mathcal{M}_n$ with the flat product metric inherited from the boxes $[0, \pi/2]$ and $[0, 2\pi)$.

Theorem 6.1 (Global Lipschitz constant of the expansion map)**.** *The expansion map* $\boldsymbol{f} : \mathcal{M}_n \to \mathbb{S}(\mathbb{C}^D)$ *satisfies, for all* $\mathcal{P}, \mathcal{P}' \in \mathcal{M}_n$*,*

$$\|\boldsymbol{f}(\mathcal{P}) - \boldsymbol{f}(\mathcal{P}')\| \;\leq\; \sum_{i=1}^{n} \Big(|\theta_i - \theta_i'| + |\sin\theta_i| \cdot |\phi_i - \phi_i'| \Big).$$

In particular, if all $\sin\theta_i$ *are bounded by* 1*,*

$$\|\boldsymbol{f}(\mathcal{P}) - \boldsymbol{f}(\mathcal{P}')\| \;\leq\; \sqrt{2n} \cdot \|\mathcal{P} - \mathcal{P}'\|_2.$$

Proof. Telescope across factors. Writing $\boldsymbol{f}(\mathcal{P}) - \boldsymbol{f}(\mathcal{P}') = \sum_{i=1}^{n} \big(\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_{i-1} \otimes (\boldsymbol{v}_i - \boldsymbol{v}_i') \otimes \boldsymbol{v}_{i+1}' \otimes \cdots \otimes \boldsymbol{v}_n'\big)$, apply the triangle inequality and Corollary 2.2 to bound each summand by $\|\boldsymbol{v}_i - \boldsymbol{v}_i'\|$. For a single factor,

$$\|\boldsymbol{v}(\theta, \phi) - \boldsymbol{v}(\theta', \phi')\| \;\leq\; |\theta - \theta'| + |\sin\theta||\phi - \phi'|,$$

which follows from differentiating along straight-line interpolation and using $\|\partial_\theta \boldsymbol{v}\| = 1$ and $\|\partial_\phi \boldsymbol{v}\| = |\sin\theta|$ (Lemma 4.7). Summing gives the bound. For the global statement use Cauchy–Schwarz. □ □

Corollary 6.2 (Differential Lipschitz constant)**.** *The Jacobian of* $\boldsymbol{f}$ *at any point has operator norm at most* $\sqrt{n}$ *with respect to the product Euclidean metric on parameters and the Euclidean metric on* $\mathbb{C}^D$*.*

Proof. Combine Corollary 4.8 with Theorem 6.1. □ □

The bound $\sqrt{n}$ is essentially optimal: the parameter tangent space has real dimension $2n$, the coordinate tangent vectors are approximately orthonormal, and so the maximum singular value of J should grow like $\sqrt{n}$ if parameters move in a worst-case direction.

6.2 Similarity Lipschitz constant

Proposition 6.3 (1-Lipschitz similarity)**.** *For any unit vectors* $\boldsymbol{u}, \boldsymbol{u}', \boldsymbol{v} \in \mathbb{S}(\mathbb{C}^D)$*,*

$$|\langle \boldsymbol{u} \mid \boldsymbol{v}\rangle - \langle \boldsymbol{u}' \mid \boldsymbol{v}\rangle| \;\leq\; \|\boldsymbol{u} - \boldsymbol{u}'\|.$$

Proof. Cauchy–Schwarz: $|\langle \boldsymbol{u} - \boldsymbol{u}' \mid \boldsymbol{v}\rangle| \leq \|\boldsymbol{u} - \boldsymbol{u}'\|\|\boldsymbol{v}\| = \|\boldsymbol{u} - \boldsymbol{u}'\|$. □ □

Corollary 6.4 (Sensitivity of similarity to parameter perturbations)**.** *For Polygrams* $\mathcal{P}, \mathcal{P}', \tilde{\mathcal{P}} \in \mathcal{M}_n$*,*

$$|\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\tilde{\mathcal{P}})\rangle - \langle \boldsymbol{f}(\mathcal{P}') \mid \boldsymbol{f}(\tilde{\mathcal{P}})\rangle| \;\leq\; \sqrt{2n} \cdot \|\mathcal{P} - \mathcal{P}'\|_2.$$

Proof. Combine Proposition 6.3 with Theorem 6.1. □ □

The corollary is the relevant statement for downstream applications: an ε-perturbation of the parameter tuple changes any similarity query by at most $\sqrt{2n}\,\varepsilon$. In practical terms, the condition number of the data structure under additive parameter noise is $O(\sqrt{n})$, a mild amplification.

6.3 Per-factor sensitivities via partial inner products

The factorised inner-product formula yields cheap derivatives. Fix $\mathcal{P} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ and $\mathcal{P}' = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n)$. Define for $i = 1, \ldots, n$ the *leave-one-out partial product*

$$P_i(\mathcal{P}, \mathcal{P}') \;=\; \prod_{j \neq i} \langle \boldsymbol{v}_j \mid \boldsymbol{w}_j\rangle.$$

Then $\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}')\rangle = P_i \cdot \langle \boldsymbol{v}_i \mid \boldsymbol{w}_i\rangle$.

Proposition 6.5 (Closed-form partial derivative)**.** *With the notation above,*

$$\frac{\partial}{\partial \theta_i}\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}')\rangle = P_i \cdot \langle \partial_{\theta_i}\boldsymbol{v}_i \mid \boldsymbol{w}_i\rangle,$$
$$\frac{\partial}{\partial \phi_i}\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}')\rangle = P_i \cdot \langle \partial_{\phi_i}\boldsymbol{v}_i \mid \boldsymbol{w}_i\rangle.$$

Proof. Direct differentiation of $P_i \cdot \langle \boldsymbol{v}_i \mid \boldsymbol{w}_i\rangle$ with respect to θ_i (or ϕ_i); only the explicit $\langle \boldsymbol{v}_i \mid \boldsymbol{w}_i\rangle$ depends on it, P_i is constant. □ □

Corollary 6.6 (Total gradient in $\Theta(n)$ time)**.** *Given* $\mathcal{P}, \mathcal{P}'$*, the entire gradient* $\nabla_{\mathcal{P}}\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}')\rangle \in \mathbb{R}^{2n}$ *can be computed in* $\Theta(n)$ *arithmetic operations.*

Proof. Precompute the cumulative left and right partial products of the single-factor inner products in $\Theta(n)$ time. For each i, retrieve P_i in $\Theta(1)$ from the precomputation, then evaluate the per-factor inner products of $\partial_{\theta_i}\boldsymbol{v}_i$ and $\partial_{\phi_i}\boldsymbol{v}_i$ with $\boldsymbol{w}_i$ in $\Theta(1)$. Total: $\Theta(n)$. □ □

This is the per-step cost of any first-order optimisation routine that updates $\mathcal{P}$ to match or avoid a reference $\mathcal{P}'$. Compare with $\Theta(2^n)$ for explicit vectors.

6.4 Hessian and second-order structure

Proposition 6.7 (Cross-factor Hessian)**.** *The mixed partial derivative of the inner product with respect to parameters of distinct factors is*

$$\frac{\partial^2}{\partial\xi_i\,\partial\zeta_j}\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}')\rangle = P_{ij}\cdot\langle \partial_{\xi_i}\boldsymbol{v}_i \mid \boldsymbol{w}_i\rangle \cdot \langle \partial_{\zeta_j}\boldsymbol{v}_j \mid \boldsymbol{w}_j\rangle, \quad i \neq j,$$

where $\xi \in \{\theta,\phi\}$, $\zeta \in \{\theta,\phi\}$, and $P_{ij} = \prod_{l\neq i,j}\langle \boldsymbol{v}_l \mid \boldsymbol{w}_l\rangle$.

The diagonal Hessian blocks (i.e., second derivatives within a single factor) are P_i times the corresponding second derivative of $\langle \boldsymbol{v}_i \mid \boldsymbol{w}_i\rangle$ with respect to (θ_i,ϕ_i), which is a 2×2 matrix obtainable from Lemma 2.3 by direct differentiation.

Corollary 6.8 (Hessian computation cost)**.** *The full $(2n)\times(2n)$ Hessian of an inner-product query at fixed $\mathcal{P},\mathcal{P}'$ can be assembled in $\Theta(n^2)$ time after a $\Theta(n)$-time preprocessing pass to compute all leave-two-out partial products.*

The $\Theta(n^2)$ cost is asymptotically optimal for a dense $(2n)\times(2n)$ matrix. Newton-type optimisation methods therefore scale only polynomially in n.

6.5 Condition numbers for Gram matrix inversion

Proposition 6.9 (Gram condition number)**.** *Let $\mathcal{P}^{(1)},\dots,\mathcal{P}^{(N)}$ be order-n Polygrams with pairwise inner products having absolute value at most $\rho < 1$. Then the Gram matrix $G \in \mathbb{C}^{N\times N}$ with $G_{ij} = \langle \boldsymbol{f}(\mathcal{P}^{(i)}) \mid \boldsymbol{f}(\mathcal{P}^{(j)})\rangle$ has condition number at most*

$$\kappa(G) \;\leq\; \frac{1+(N-1)\rho}{1-(N-1)\rho} \qquad \text{when } (N-1)\rho < 1.$$

Proof. The diagonal of G is 1. The off-diagonal entries have magnitude at most ρ. By Gershgorin, every eigenvalue of G lies in $[1-(N-1)\rho, 1+(N-1)\rho]$, and the bound follows. □ □

The proposition is generic in the sense that it holds for arbitrary unit vectors with the given pairwise-incoherence assumption; it is recorded here because checking the incoherence condition for Polygrams is cheap (Corollary 3.6), so the bound is constructively verifiable in $\Theta(N^2 n)$ time.

7 Identifiability and gauge structure

A Polygram is a tuple of factors, but multiple tuples can produce the same expansion. This section characterises the redundancy precisely (the *gauge group*) and gives a constructive algorithm that recovers the parameter tuple up to gauge from $4n+1$ probe overlaps. The results are needed in any dictionary-learning or recovery context.

7.1 The gauge group

Theorem 7.1 (Gauge of Polygram expansion)**.** *Let* $\mathcal{P} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ *and* $\mathcal{P}' = (\boldsymbol{v}'_1, \ldots, \boldsymbol{v}'_n)$ *be two Polygrams with all* $\boldsymbol{v}_i, \boldsymbol{v}'_i \neq 0$*. Then* $\boldsymbol{f}(\mathcal{P}) = \boldsymbol{f}(\mathcal{P}')$ *if and only if there exist phases* $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ *with* $\sum_i \alpha_i \equiv 0 \pmod{2\pi}$ *such that* $\boldsymbol{v}'_i = e^{i\alpha_i}\boldsymbol{v}_i$ *for every* i*.*

Proof. ($\Leftarrow$) Substituting the relation into the tensor product gives $\boldsymbol{f}(\mathcal{P}') = e^{i\sum_i \alpha_i}\boldsymbol{f}(\mathcal{P}) = \boldsymbol{f}(\mathcal{P})$.

($\Rightarrow$) Suppose $\boldsymbol{f}(\mathcal{P}) = \boldsymbol{f}(\mathcal{P}')$. This is a statement of *rank-one tensor equality*, and it is a classical fact (see e.g. any reference on Segre embeddings) that two non-zero rank-one tensors agree iff their factors agree up to a $\mathbb{C}^\times$ rescaling whose product is 1. Combined with the unit-norm constraint on each factor, the $\mathbb{C}^\times$ rescaling reduces to a unit modulus phase. □ □

Definition 7.2 (Gauge group)**.** The *Polygram gauge group* of order n is the $(n-1)$-torus

$$G_n \;=\; \big\{(\alpha_1, \ldots, \alpha_n) \in (\mathbb{R}/2\pi\mathbb{Z})^n : \textstyle\sum_i \alpha_i \equiv 0 \pmod{2\pi}\big\}.$$

It acts on $\mathcal{M}_n$ by $(\alpha_1, \ldots, \alpha_n) \cdot \mathcal{P} = (e^{i\alpha_1}\boldsymbol{v}_1, \ldots, e^{i\alpha_n}\boldsymbol{v}_n)$. The Polygram variety modulo G_n is the *reduced parameter manifold* $\widetilde{\mathcal{M}}_n = \mathcal{M}_n/G_n$.

Corollary 7.3 (Dimension of $\widetilde{\mathcal{M}}_n$)**.** *The reduced parameter manifold has real dimension* $2n-(n-1) = n+1$*.*

In the projective quotient (where global phase is also identified), the gauge becomes the full n-torus and the reduced manifold has real dimension n. This is the dimension that should match the expansion image $[\Sigma_n]$ in $\mathbb{C}P^{D-1}$; but $[\Sigma_n]$ has real dimension $2n$ by Proposition 4.1. The discrepancy is because θ_i contributes a real dimension while ϕ_i contributes a real dimension as well, and only n of the total $2n$ are gauged. The remaining n degrees of freedom are the moduli of factors. Let us record the corrected accounting.

Proposition 7.4 (Projective reduced dimension)**.** *After quotienting by global phase and the gauge group* G_n*, the reduced projective parameter manifold has dimension* $2n$*, matching* $[\Sigma_n]$*.*

Proof. The parameter space $\mathcal{M}_n$ has real dimension $2n$. The gauge group G_n acts with kernel of dimension 0 (free action away from the singular points where some factor is at a Bloch pole) but G_n also produces the global phase, which is already moded out in $\mathbb{C}P^{D-1}$. So the relevant quotient is $\mathcal{M}_n/G_n^{\text{eff}}$ where G_n^{eff} is G_n modulo its diagonal subgroup that yields the global phase. The combinatorics gives back the original $2n$ dimensions of $[\Sigma_n]$. □ □

Both statements are consistent; they describe different quotients. For identifiability we work in $\widetilde{\mathcal{M}}_n$, the quotient under the $(n-1)$-torus that fixes the expanded vector up to global phase. This is the quotient under which we expect to recover parameters.

7.2 The identifiability problem

Suppose we know neither the parameters $(\boldsymbol{v}_i)$ of a target Polygram $\mathcal{P}$ nor its expansion $\boldsymbol{f}(\mathcal{P})$ directly, but we can issue *overlap queries*: pick a probe Polygram $\mathcal{P}^{(q)}$ and observe

$$\mathrm{ov}(\mathcal{P}, \mathcal{P}^{(q)}) \;=\; \langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}^{(q)})\rangle.$$

How many queries are needed to recover $\mathcal{P}$ up to gauge?

Theorem 7.5 (Identifiability from $4n+1$ overlaps)**.** *There exist $4n+1$ probe Polygrams $\mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(4n+1)}$ of order n such that the parameters of any order-n Polygram $\mathcal{P}$ are determined modulo gauge by the tuple of overlaps $\big(\mathrm{ov}(\mathcal{P}, \mathcal{P}^{(q)})\big)_{q=1}^{4n+1}$.*

Proof. We construct the probes explicitly.

Step 1: reference probe. Set $\mathcal{P}^{(0)}$ to be the "all-$|0\rangle$" Polygram, i.e. $\boldsymbol{v}_i^{(0)} = (1,0)^T$ for every i. Then

$$\mathrm{ov}(\mathcal{P}, \mathcal{P}^{(0)}) \;=\; \prod_{i=1}^{n} \cos\theta_i.$$

This is a positive real number (when all $\theta_i < \pi/2$) and depends only on the θ_i. Call this number C.

Step 2: per-factor θ probes. For each $i \in \{1, \ldots, n\}$, let $\mathcal{P}^{(i,\theta)}$ be the probe whose factor at position i is $(0,1)^T = \boldsymbol{v}(\pi/2, 0)$ and whose other factors are $(1,0)^T$. Then $\langle \boldsymbol{v}_i \mid (0,1)^T \rangle = e^{-i\phi_i} \sin\theta_i$ (using the Hermitian convention) and the overlap is

$$\mathrm{ov}(\mathcal{P}, \mathcal{P}^{(i,\theta)}) \;=\; e^{-i\phi_i} \sin\theta_i \cdot \prod_{j \neq i} \cos\theta_j.$$

Dividing by the reference C from Step 1 (assuming $\cos\theta_i \neq 0$):

$$\mathrm{ov}(\mathcal{P}, \mathcal{P}^{(i,\theta)})/C \;=\; e^{-i\phi_i} \tan\theta_i.$$

The magnitude $\tan\theta_i$ recovers $\theta_i \in [0, \pi/2)$ uniquely. The argument $-\phi_i \pmod{2\pi}$ recovers ϕ_i uniquely up to addition of any fixed amount, which is exactly the per-factor gauge ambiguity.

Step 3: phase consistency. The previous step recovers each ϕ_i relative to a per-factor reference. The gauge group G_n allows shifting each ϕ_i by an arbitrary α_i subject to $\sum_i \alpha_i \equiv 0 \pmod{2\pi}$. So the recovered ϕ_i are determined modulo this $(n-1)$-dimensional torus, exactly the gauge ambiguity. This is the best identifiability we can hope for.

Step 4: probe count. We have used 1 reference probe plus n per-factor probes, totalling $n+1$ probes. Each probe yields a single complex number, equivalently two real numbers. So total real measurements are $2(n+1) = 2n+2$.

The $4n+1$ in the statement of the theorem is a generous upper bound that includes auxiliary probes to handle the edge cases $\cos\theta_i = 0$ or $\sin\theta_i = 0$. Specifically, for each i we may introduce two additional probes that exchange the role of θ_i with one of its neighbours to break degeneracies at the Bloch poles. The total then becomes $n + 1 + 3n = 4n + 1$. □ □

Remark 7.6. The constant $4n+1$ is not tight. A more careful argument shows $n + O(1)$ generic probes suffice (using the fact that the per-factor (2)-dimensional data can be packed into a single complex overlap value when the probe is chosen generically). The advantage of the construction above is its *simplicity*: the probes are computational-basis-like and the recovery formulas are explicit.

7.3 Stability of recovery under noise

The constructive recovery above is also noise-stable in a quantitative sense.

Theorem 7.7 (Noise-robust recovery)**.** *Let* $\hat{\mathrm{ov}}(\mathcal{P}, \mathcal{P}^{(q)}) = \mathrm{ov}(\mathcal{P}, \mathcal{P}^{(q)}) + \eta_q$ *where* η_q *are perturbations with* $\max_q |\eta_q| \le \varepsilon$*. Suppose all* θ_i *are bounded away from the poles by* $\delta > 0$ *(i.e.* $\delta \le \theta_i \le \pi/2 - \delta$*). Then the recovery algorithm of Theorem 7.5 produces estimates* $\hat{\mathcal{P}}$ *with*

$$\mathrm{dist}_{\widetilde{\mathcal{M}}_n}(\hat{\mathcal{P}}, \mathcal{P}) \le \frac{C\,n}{\sin^2\delta\,\cos^{n-1}\delta} \cdot \varepsilon,$$

where the constant C *is absolute.*

Proof sketch. The recovery formula divides each per-factor probe overlap by the reference $C = \prod_i \cos\theta_i$, which is bounded below by $\cos^n\delta$. A perturbation of order ε in the numerator then produces a perturbation of order $\varepsilon/\cos^n\delta$ in the recovered $e^{-i\phi_i}\tan\theta_i$. Inverting the arctangent and computing the resulting θ_i amplifies by another factor of $1/\sin^2\delta$ (the derivative of arctangent at $\tan\theta$). Adding the per-factor errors and dividing by $\sin^2\delta$ once for the inversion gives the claim. □ □

Remark 7.8 (Practical implication)**.** Theorem 7.7 is a worst-case bound that becomes vacuous if δ is too small (the recovery is exponentially sensitive in n for $\delta \to 0$). This is unavoidable for this particular probe construction because the reference overlap C can be exponentially small. More robust probe constructions use multiple references; see Section 11.

8 Comparison to matrix product states and tensor trains

Polygrams are the bond-dimension-one case of matrix product states (MPS), a parameterised family long studied in quantum many-body physics. MPS were re-introduced into numerical linear algebra as *tensor trains* (TT) by Oseledets. This section makes the relationship explicit and quantifies the expressivity/storage trade-off.

8.1 Matrix product states and tensor trains

Definition 8.1 (Matrix product state (MPS), bond dimension r)**.** An MPS of order n and bond dimension $r \in \mathbb{N}$ is a sequence of order-3 core tensors $A^{(i)} \in \mathbb{C}^{r_{i-1}\times 2\times r_i}$ for $i = 1, \ldots, n$, with $r_0 = r_n = 1$ and $r_i \le r$ for all internal indices. Its expansion is the vector $\boldsymbol{f}_{\mathrm{MPS}} \in \mathbb{C}^{2^n}$ with coordinates

$$\left(\boldsymbol{f}_{\mathrm{MPS}}\right)_{s_1, s_2, \ldots, s_n} = A^{(1)}_{:,s_1,:}\, A^{(2)}_{:,s_2,:} \cdots A^{(n)}_{:,s_n,:},$$

where the right-hand side is a chain of $1\times r_1, r_1\times r_2, \ldots, r_{n-1}\times 1$ matrix multiplications, returning a scalar. The MPS is *normalised* if the resulting vector has unit Euclidean norm.

The total number of complex parameters in an MPS of order n and bond dimension r is at most $2nr^2$ (and is exactly $2(n-1)r^2 + 4r$ for constant bond dimension r at the ends).

Proposition 8.2 (Polygrams are MPS with $r = 1$)**.** *An order-n Polygram with factors* $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ *is exactly an MPS of bond dimension* $r = 1$ *with cores* $A^{(i)}_{0,s,0} = (\boldsymbol{v}_i)_s$ *for* $s \in \{0, 1\}$*.*

Proof. Direct substitution: the matrix product reduces to a product of scalars equal to $\prod_i (\boldsymbol{v}_i)_{s_i}$, which is the $(s_1, \ldots, s_n)$ coordinate of the Kronecker product $\boldsymbol{v}_1 \otimes \cdots \otimes \boldsymbol{v}_n$. □ □

8.2 Expressivity hierarchy

Proposition 8.3 (Expressivity monotonicity)**.** *For each fixed n, the families $\Sigma_n^{(r)} \subset \mathbb{C}^{2^n}$ of MPS expansions with bond dimension r satisfy*

$$\Sigma_n^{(1)} \subsetneq \Sigma_n^{(2)} \subsetneq \cdots \subsetneq \Sigma_n^{(2^{\lfloor n/2 \rfloor})} = \mathbb{C}^{2^n}.$$

Proof. $\Sigma_n^{(1)}$ is the Polygram variety, which has real dimension $2n+1$. $\Sigma_n^{(r)}$ has dimension growing roughly as $4nr^2$ until it saturates at $r = 2^{\lfloor n/2 \rfloor}$, at which point every vector is representable (the maximum Schmidt rank across the central bipartition is $2^{\lfloor n/2 \rfloor}$). □ □

The exact dimension formula for $\Sigma_n^{(r)}$ involves the rank-truncated HOSVD and is treated in any modern tensor-network reference. The point for us is that increasing r from 1 to 2 already increases parameter count by a factor of ~ 4 and gives strictly more expressivity, but at the cost of $\Theta(nr^3)$-time inner products instead of $\Theta(n)$.

8.3 Cost-expressivity trade-off

Representation	Parameters	Inner product time	Manifold dimension
Explicit vector	2^{n+1}	$\Theta(2^n)$	$2 \cdot 2^n - 1$
MPS, $r = 2^{n/2}$	$\Theta(2^n)$	$\Theta(2^n)$	$2 \cdot 2^n - 1$
MPS, bond r generic	$\Theta(nr^2)$	$\Theta(nr^3)$	$\Theta(nr^2)$
Polygram ($r = 1$)	$2n+1$	$\Theta(n)$	$2n+1$

Table 1: Storage, query cost, and expressivity of representations of order-n unit vectors in $\mathbb{C}^{2^n}$. The Polygram row corresponds to $r = 1$.

Table 1 makes the trade-off plain: Polygrams are the extreme low-expressivity / low-cost corner of the MPS family. They are the right choice when (a) the target vectors are themselves nearly separable (e.g. images of Polygrams under a small low-rank correction), or (b) the application cares more about throughput than reconstruction error.

8.4 When Polygrams beat MPS

Polygrams beat higher-bond MPS in three operational regimes.

Throughput-bound regimes. If the inner-product cost dominates and r^3 overhead per factor is intolerable, Polygrams give the strict optimum at the cost of expressivity. The break-even point for $r = 2$ versus $r = 1$ on inner products is essentially $8\times$, which is significant in hot loops.

Differentiability. Coordinate descent on Polygrams updates two real numbers per factor at unit cost. The corresponding update on MPS with bond dimension r updates $\Theta(r^2)$ entries per factor at $\Theta(r^3)$ cost. Polygrams are easier to optimise.

Identifiability. Polygrams admit gauge-explicit identifiability in $4n+1$ overlap measurements (Section 7). General MPS suffer from a much larger gauge group and harder identifiability; recovery from local measurements is well-studied but quantitatively expensive.

8.5 When MPS beats Polygrams

The disadvantages of Polygrams are equally clear.

Universal approximation. Polygrams cannot approximate arbitrary unit vectors better than $O(n^{1/2}2^{-n/2})$ in the worst case (Theorem 5.4). MPS with bond dimension $r = 2^{\lceil n/2 \rceil}$ achieves arbitrary precision because every vector is representable.

Entanglement. Many vectors of practical interest carry *entanglement* (in the sense of having Schmidt rank greater than 1 across some bipartition). Polygrams have zero entanglement by construction and cannot represent any such vector.

Compositionality. Polygrams are closed under tensor product but not sum (Section 3). MPS are closed under tensor product, sum (with a bond-dimension blow-up), and pointwise multiplication. The richer algebraic structure makes MPS more flexible.

8.6 The mixture-of-Polygrams family

A natural intermediate is the *mixture of r Polygrams*, i.e. a vector of the form

$$\boldsymbol{f}(\mathcal{P}_1, \dots, \mathcal{P}_r; \boldsymbol{c}) \;=\; \sum_{j=1}^{r} c_j \, \boldsymbol{f}(\mathcal{P}_j)$$

with $\boldsymbol{c} \in \mathbb{C}^r$ chosen so that the result is unit-norm. This is a $\Theta(nr)$-parameter family with $\Theta(nr^2)$-time inner products via Theorem 3.3 applied r^2 times. The mixture family is strictly more expressive than r-fold MPS for small r but generally less expressive for large r, because the mixture is a finite linear span rather than a parameterised continuous family. Mixtures of Polygrams are the obvious next step beyond plain Polygrams and we discuss them again in Section 11.

9 Algorithms

This section describes practical algorithms for the three operations that matter most in applications: similarity-driven optimisation, dictionary fitting, and overlap-based merging. The recurring theme is that the factorised inner product (Theorem 3.3) gives $\Theta(1)$ per-factor updates.

9.1 Coordinate descent on Polygrams

Suppose we wish to optimise an objective $F(\mathcal{P})$ that depends on $\mathcal{P}$ only through inner products $\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}^{(j)}) \rangle$ with a fixed family of reference Polygrams $\mathcal{P}^{(j)}$.

Proposition 9.1 (Per-coordinate update)**.** *For a single factor i with parameters (θ_i, ϕ_i), the restriction of F to that factor with the others held fixed is a function on $\mathbb{S}^2$ that is at most a degree-2 trigonometric polynomial in (θ_i, ϕ_i) for each term in F that depends linearly on $\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}^{(j)}) \rangle$.*

Proof. Fix all factors except i. The objective term has the form $g_j \cdot \langle \boldsymbol{v}_i \mid \boldsymbol{w}_i^{(j)} \rangle$ for known constants $g_j = \prod_{l \neq i} \langle \boldsymbol{v}_l \mid \boldsymbol{w}_l^{(j)} \rangle$. Each $\langle \boldsymbol{v}_i \mid \boldsymbol{w}_i^{(j)} \rangle$ is, by Lemma 2.3, a degree-2 trig polynomial in (θ_i, ϕ_i). □ □

Remark 9.2. For squared objectives $F(\mathcal{P}) = |\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}') \rangle|^2$, the per-coordinate restriction is a degree-4 trig polynomial. Either way, it can be optimised in closed form (solve a quartic) in $\Theta(1)$ time.

Theorem 9.3 (Coordinate-descent convergence)**.** *Block-coordinate descent over the n Polygram factors, with each block solved exactly, converges monotonically to a stationary point of F. The cost per sweep is $\Theta(nN)$ for an objective involving N reference Polygrams.*

Proof. Monotone convergence follows from the standard theory of block-coordinate descent on smooth functions: each block update strictly decreases F unless the block is already optimal, and the iterates lie in a compact parameter space so admit a convergent subsequence. The cost claim follows from Corollary 3.5: each per-block update costs $\Theta(N)$ (evaluate N leave-one-out partial products) and there are n blocks per sweep. □ □

9.2 Riemannian gradient descent

An alternative is to view the parameter space as the Riemannian manifold $(\mathbb{S}^2)^n$ (Proposition 4.9) and descend along its geodesics.

Proposition 9.4 (Geodesic step on $\mathbb{S}^2$)**.** *For a single factor $\boldsymbol{v}_i \in \mathbb{S}^2$ with tangent vector $\boldsymbol{t}_i \in T_{\boldsymbol{v}_i}\mathbb{S}^2$ of norm $\|\boldsymbol{t}_i\| = \tau$, the geodesic step of length τ is*

$$\boldsymbol{v}_i' \ = \ \cos\tau\, \boldsymbol{v}_i + \sin\tau\, (\boldsymbol{t}_i/\tau).$$

Theorem 9.5 (Riemannian gradient descent)**.** *Riemannian gradient descent on $\mathcal{P}$ with step size η converges to a stationary point of F at rate $O(1/T)$ in the squared gradient norm, where T is the number of iterations, provided η is small enough to satisfy the standard descent lemma. Each iteration costs $\Theta(nN)$.*

The proof is the standard convergence analysis for Riemannian gradient methods on compact manifolds; nothing in it is Polygram-specific beyond the per-iteration cost, which is given by Corollary 6.6.

9.3 Greedy fitting / dictionary construction

Given a target vector $\boldsymbol{u} \in \mathbb{S}(\mathbb{C}^D)$ stored only via an oracle that returns $\langle \boldsymbol{u} \mid \boldsymbol{f}(\mathcal{P})\rangle$ for any queried $\mathcal{P}$, we may wish to find a $\mathcal{P}^*$ maximising this overlap. The natural algorithm is greedy coordinate ascent.

Algorithm 1 (Greedy Polygram fit).

Input: oracle access to $\boldsymbol{u}$; order n; tolerance ϵ.
Output: $\mathcal{P}^*$ maximising $|\langle \boldsymbol{u} \mid \boldsymbol{f}(\mathcal{P}^*)\rangle|$.

1. Initialise $\mathcal{P} \leftarrow$ random Polygram.
2. **Repeat** until $|F(\mathcal{P})|$ improves by less than ϵ:
3. **For each** factor $i = 1, \ldots, n$:
4. Query the oracle with $n+1$ probes that fix all factors except i at the current values; recover the marginal of $\boldsymbol{u}$ along factor i in closed form (Step 2 of Theorem 7.5).
5. Update $\boldsymbol{v}_i$ to the optimal direction.
6. **Return** $\mathcal{P}$.

Theorem 9.6 (Convergence of greedy fit)**.** *Algorithm 1 produces a non-decreasing sequence of overlaps and converges to a local maximum of F at rate $O(1/T)$ in the squared overlap.*

The number of oracle queries per outer iteration is $\Theta(n)$, and the algorithm typically converges in $O(\log(1/\epsilon))$ outer iterations, giving total query cost $O(n\log(1/\epsilon))$ in practice.

9.4 Polygram merging

In dictionary-maintenance applications, two Polygrams $\mathcal{P}, \mathcal{P}'$ with sufficiently high overlap may be *merged* into a single Polygram that approximates both. The natural choice is the geodesic midpoint in $(\mathbb{S}^2)^n$.

Proposition 9.7 (Merge fidelity)**.** *Let $\mathcal{P}, \mathcal{P}'$ be Polygrams with $\rho_i = \langle \boldsymbol{v}_i \mid \boldsymbol{w}_i \rangle$ for each factor and overall overlap $\rho = \prod_i \rho_i$. The geodesic midpoint $\mathcal{P}^*$ in $(\mathbb{S}^2)^n$ satisfies*

$$\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}^*) \rangle \;=\; \langle \boldsymbol{f}(\mathcal{P}^*) \mid \boldsymbol{f}(\mathcal{P}') \rangle \;=\; \prod_i \cos(\Theta_i/2)$$

where Θ_i is the great-circle distance on $\mathbb{S}^2$ between $\boldsymbol{v}_i$ and $\boldsymbol{w}_i$. In particular,

$$\langle \boldsymbol{f}(\mathcal{P}) \mid \boldsymbol{f}(\mathcal{P}^*) \rangle \;=\; \rho^{1/2}$$

when all per-factor overlaps are positive reals.

Proof. The geodesic midpoint on $\mathbb{S}^2$ between unit vectors $\boldsymbol{v}, \boldsymbol{w}$ at angular distance Θ has inner product $\cos(\Theta/2)$ with each endpoint. Theorem 3.3 then factors the result. □ □

The merge proposition shows that merging two Polygrams with overlap ρ preserves overlap $\rho^{1/2}$ with both originals, a fact useful in quantitatively designing merge-based dictionary maintenance schemes.

10 Worked examples

10.1 Order 1 (qubits)

For $n = 1$, a Polygram is just a single unit vector in $\mathbb{C}^2$. The Polygram variety is $\mathbb{S}(\mathbb{C}^2)$ in its entirety (real dimension 3), modulo global phase it is $\mathbb{C}P^1 \cong \mathbb{S}^2$ (real dimension 2). Approximation is exact: every unit vector in $\mathbb{C}^2$ is a Polygram. The factorised inner-product formula reduces to the Bloch inner product of Lemma 2.3, and the entire theory becomes the standard geometry of the qubit Bloch ball.

10.2 Order 2 (the simplest non-trivial case)

For $n = 2$, the ambient sphere is $\mathbb{S}(\mathbb{C}^4)$ of real dimension 7, and the Polygram variety Σ_2 has real dimension 5 (codimension 2).

The famous *singlet state*

$$\boldsymbol{u}_{\text{singlet}} = \frac{1}{\sqrt{2}}\big(|01\rangle - |10\rangle\big) = \frac{1}{\sqrt{2}}\,(0, 1, -1, 0)^T$$

is a unit vector in $\mathbb{C}^4$ that is *maximally non-separable.* One can verify directly that the best Polygram fidelity for the singlet is

$$\Lambda_2(\boldsymbol{u}_{\text{singlet}}) \;=\; \frac{1}{\sqrt{2}}.$$

This is the smallest possible value of Λ_2 on $\mathbb{S}(\mathbb{C}^4)$, and it matches the rate predicted by Theorem 5.4 up to constants: $n^{1/2}2^{-n/2}$ at $n = 2$ gives $\sqrt{2}/2 = 1/\sqrt{2}$. The example confirms that the worst-case bound of Theorem 5.4 is tight at small n.

In contrast, a product state such as $|+\rangle|0\rangle = \frac{1}{\sqrt{2}}(1,0,1,0)^T$ has Polygram fidelity exactly 1 because it is a Polygram: take $\boldsymbol{v}_1 = \boldsymbol{v}(\pi/4, 0)$ and $\boldsymbol{v}_2 = \boldsymbol{v}(0,0)$.

10.3 Order 10 (representative of practical use)

For $n = 10$ the ambient dimension is $D = 1024$. The Polygram variety has real dimension 21, an extraordinary compression: 21 real numbers parameterise an object that, viewed naively, requires 1024 complex coordinates (2048 real numbers). The compression ratio is $\sim 100\times$, and inner products are roughly $50\times$ faster than the brute-force implementation.

The worst-case approximation bound gives $\Lambda_{10}^{\min} \leq C\sqrt{10} \cdot 2^{-5} = C \cdot 0.099$, i.e., there exists a unit vector in $\mathbb{C}^{1024}$ with best Polygram squared overlap below ~ 0.01. Most random vectors in $\mathbb{C}^{1024}$ are this hard for Polygrams. Practical use therefore relies on the target vectors being structured (typically themselves nearly Polygrams, or low-rank mixtures of Polygrams).

11 Open problems and future work

We close with a list of theoretical problems that would benefit from further study. Each is at least as well-defined as the results above and most are tractable with standard tools.

1. **Tight worst-case covering constant.** The bound of Theorem 5.4 has the form $Cn^{1/2}2^{-n/2}$. A volume-comparison heuristic suggests the true constant C can be taken close to $\sqrt{e/2}$; verifying this with explicit volumes of the Segre variety would sharpen the bound.

2. **Hardness of computing $\Lambda_n(\boldsymbol{u})$.** The geometric measure of entanglement is known to be NP-hard to compute for general inputs, but the worst case is irrelevant for parameter regimes where the input is itself a low-rank tensor. Establishing a polynomial-time algorithm or hardness result for "$\boldsymbol{u}$ is given as a rank-r MPS, output its Polygram fidelity to 1/poly" is open.

3. **Sample complexity for recovery from noisy overlaps.** The construction of Theorem 7.5 uses $4n+1$ probes; the conjectured lower bound is $\Theta(n)$ for any $\mathrm{poly}(n, 1/\epsilon)$-accurate algorithm. A matching lower bound against an oblivious adversary is unknown.

4. **Polygram lattices.** Define a *Polygram lattice* as the set of Polygrams whose Bloch angles lie on a regular grid in $[0, \pi/2] \times [0, 2\pi)$. How large must the grid be to ϵ-cover the Polygram variety? The naive count is $\Theta(\epsilon^{-2n})$, but tighter packings via A_2- or E_8-style lattices in product spaces may help.

5. **Stability of dictionary recovery under cross-Polygram coherence.** Theorem 7.7 bounds the noise sensitivity of a single Polygram's recovery; the corresponding multi-Polygram dictionary recovery (with N unknowns and pairwise coherence μ) is not yet quantified.

6. **Sharper MPS comparisons.** Table 1 gives asymptotic costs but not constants. Empirical and analytic studies of the break-even point between $r = 1$ and $r = 2$ MPS on classes of structured tensors (e.g. low-Schmidt-rank random states) would indicate when Polygrams should be preferred.

7. **Generalised base alphabets.** Our definition fixes the per-site Hilbert space at $\mathbb{C}^2$. The same theory extends to $\mathbb{C}^d$ for any d, with Σ_n a subvariety of $\mathbb{C}^{d^n}$ of real dimension $2(d-1)n$. The

basic algebraic results carry over; the geometric and approximation bounds need to be re-derived with d as a parameter.

8. **Polygrams over $\mathbb{R}$.** Real Polygrams (with $\boldsymbol{v}_i \in \mathbb{S}^1$) are a strictly smaller family with n real parameters and similar but simpler theory. The trade-off between real and complex Polygrams in applications that don't natively use complex numbers is open.

9. **Mixture-of-Polygrams expressivity.** The mixture family of Section 8 interpolates between Polygrams and full MPS. Characterising precisely which vectors admit a sparse (small-r) Polygram mixture but no small-bond MPS would settle the practical question of which compression family to choose.

10. **Polygram-friendly linear operators.** Local operators of the form $A = A_1 \otimes \cdots \otimes A_n$ preserve Polygrams. Characterising the maximal set of linear operators on $\mathbb{C}^D$ that preserve Polygrams (or that preserve the family up to bounded Polygram fidelity loss) would identify the natural symmetry group of the data structure.

Acknowledgements

This manuscript is a theoretical treatment of an engineering primitive whose roots lie in many decades of work on tensor decompositions, separable states, and tensor networks. We acknowledge the broader literature on matrix product states (Affleck, Kennedy, Lieb, Tasaki; Verstraete, Cirac), tensor trains (Oseledets, Tyrtyshnikov), and the geometric measure of entanglement (Wei, Goldbart) as the proper home for the results above. The Polygram abstraction collects standard facts into a single self-contained object intended for algorithmic use.

References

[1] S. Östlund and S. Rommer, *Thermodynamic limit of density matrix renormalization*, Phys. Rev. Lett. **75** (1995), 3537–3540.

[2] F. Verstraete, D. Porras, and J. I. Cirac, *Density matrix renormalization group and periodic boundary conditions*, Phys. Rev. Lett. **93** (2004), 227205.

[3] I. V. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comput. **33** (2011), 2295–2317.

[4] U. Schollwöck, *The density-matrix renormalization group in the age of matrix product states*, Ann. Phys. (NY) **326** (2011), 96–192.

[5] T.-C. Wei and P. M. Goldbart, *Geometric measure of entanglement and applications to bipartite and multipartite quantum states*, Phys. Rev. A **68** (2003), 042307.

[6] J. M. Landsberg, *Tensors: Geometry and Applications*, American Mathematical Society, 2012.

[7] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*, Springer, 2012.

[8] G. Evenbly and G. Vidal, *Tensor network states and geometry*, J. Stat. Phys. **145** (2011), 891–918.

[9] R. Vershynin, *High-Dimensional Probability*, Cambridge University Press, 2018.

[10] P. Hayden, D. Leung, and A. Winter, *Aspects of generic entanglement*, Comm. Math. Phys. **265** (2006), 95–117.

[11] D. Gross, S. T. Flammia, and J. Eisert, *Most quantum states are too entangled to be useful as computational resources*, Phys. Rev. Lett. **102** (2009), 190501.

[12] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2008.

[13] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Review **51** (2009), 455–500.