# AXIOM-TENSOR:

## Diagonally Modulated Signed-XOR Operators

### Representation, Initialization, Optimization Geometry, and Exact Execution

Park Junhu

Interdisciplinary Program in Computational Science and Technology
Seoul National University, Seoul, Republic of Korea
nller.park@snu.ac.kr

**Abstract**

Representation, initialization, optimization, and execution are commonly analyzed separately for structured linear operators. We study these questions as distinct views of one parameter-to-operator map, $\mathcal{F}_R(A, U, V) = \sum_r \operatorname{diag}(U_r) S_\phi(A_r) \operatorname{diag}(V_r)$, where $S_\phi$ reads a profile through a fixed signed index map and $R$ is the component count. AXIOM-TENSOR is the signed-XOR specialization $\phi(i, j) = i \oplus j$ induced by Cayley–Dickson basis multiplication.

For representation, a sign-corrected XOR-slice map is a Frobenius isometry and one component can prescribe an arbitrary XOR diagonal. This gives an exact $n$-component compiler for every $n \times n$ matrix. The construction is not, in general, a compression: whenever $3Rn < n^2$, the fixed-$R$ image has measure zero in matrix space. For initialization, a Latin-square condition yields the unique variance-preserving coefficient scale $1/\sqrt{Rn}$, called ARCANA (Algebraic Rank-Corrected Activation-Normalized Amplitude), together with a homogeneous-activation moment correction. Variance preservation is a step-zero moment statement, not a convergence result. For optimization, the differential shows that symmetric component copies have identical profile gradients; the resulting first-order operator update ranges from $\sqrt{R}$ to $R$ scaling according to copy coherence. This is a local geometric statement, not a universal learning-rate rule. For execution, the fixed coefficients admit an exact signed-permutation graph with $3Rn$ learned scalars. Exact execution is relative to the selected AXIOM operator and implies neither exact recovery of the original dense target nor a speedup.

The images are nested as the component count increases, so best approximation error is nonincreasing. This is the mathematical basis for adaptive representation; no runtime selection policy is validated here. A binary-carry theorem further shows that fixed local stencils occupy only $O(\log n)$ XOR diagonals, although a single component may have full matrix rank. Minimal checks confirm the predicted Gaussian projection scale, the XOR-support mechanism for full-rank stencils, and compatibility with training a small MNIST model. These checks are not evidence for universal dense-matrix compression.

**Keywords:** structured matrices; Cayley–Dickson algebra; signed XOR; ARCANA initialization; optimization geometry; exact compilation; structured matrix approximation.

## 1 Introduction

Structured linear operators are studied through several questions that are usually separated by purpose. Representation theory asks which matrices a parameterization can attain. Initialization analysis asks how random coefficients propagate second moments. Optimization analysis concerns the differential from parameter space to operator space. Execution analysis asks whether fixed coefficients can be evaluated without dense materialization. Finally, varying the parameter budget produces a nested

approximation family. These questions require different mathematical objects, even when they arise from the same parameterization.

We place them in one notation for component-summed shared-profile operators. Let $\phi(i, j)$ be a fixed index map and let $\vartheta_{ij}$ be a fixed unit-modulus gain. A profile $a$ defines

$$S_\phi(a)_{ij} = \vartheta_{ij} a_{\phi(i,j)}, \qquad \mathcal{F}_R(A, U, V) = \sum_{r=1}^{R} \operatorname{diag}(U_r) S_\phi(A_r) \operatorname{diag}(V_r). \tag{1.1}$$

Here and throughout, $R$ is the component count, not the ordinary matrix rank. The same map gives five mathematical views:

| **View** | Mathematical object | Question |
|---|---|---|
| **Representation** | image $\operatorname{Im} \mathcal{F}_R$ | Which matrices occur at a fixed component count? |
| **Initialization** | pushforward covariance | Which coefficient scale preserves signal energy? |
| **Optimization geometry** | differential $D\mathcal{F}_R$ | How do parameter updates combine in operator space? |
| **Exact execution** | factorized arithmetic graph | Does the graph evaluate the selected operator exactly? |
| **Adaptive representation** | nested images $\{\operatorname{Im} \mathcal{F}_R\}_R$ | How does error vary with component count? |

The views are related but not interchangeable. Variance preservation does not imply convergence; exact representation does not imply compression; an exact compiler need not produce a smaller representation; and exact execution of a selected structured operator implies neither exact recovery of a dense target nor a hardware speedup. Likewise, nesting of the images is a monotonicity statement, not validation of a runtime controller.

At the symmetric initialization $U_r = V_r = \mathbf{1}$, linearity gives

$$\mathcal{F}_R(A, \mathbf{1}, \mathbf{1}) = S_\phi\left(\sum_{r=1}^{R} A_r\right). \tag{1.2}$$

Independent components initialized at the single-component scale would give $R$ times the intended variance. Dividing the coefficient standard deviation by $\sqrt{R}$ gives the per-component amplitude called ARCANA, short for *Algebraic Rank-Corrected Activation-Normalized Amplitude*. The same identity makes the symmetric profile derivatives equal, so the first operator update can still depend on component count. Initialization and optimization geometry are therefore two consequences of Eq. (1.2), but they concern different statistical objects.

AXIOM-TENSOR specializes Eq. (1.1) to $n = 2^k$, $\phi(i, j) = i \oplus j$, and $\vartheta_{ij} = \theta(i \oplus j, j)$, where

$$e_p e_q = \theta(p, q) e_{p \oplus q}, \qquad \theta(p, q) \in \{-1, +1\}. \tag{1.3}$$

Reindexing a matrix by $p = i \oplus j$ and removing the fixed sign gives

$$X\big(\mathcal{F}_R(A, U, V)\big)_{p,i} = \sum_{r=1}^{R} A_{rp} U_{ri} V_{r,i \oplus p}. \tag{1.4}$$

This coordinate form separates component count from matrix rank: one component can have full matrix rank and can prescribe one XOR diagonal, whereas multiple diagonals remain coupled through shared row and permuted-column profiles.

Section 2 positions the operator among related structured matrix families. Section 3 defines the general map and its signed-XOR specialization. Sections 4–8 develop representation, initialization, optimization geometry, exact execution, and the nested-image consequence. Sections 9 and 10 give minimal numerical checks. Sections 11 and 12 separate the proved statements from their scope conditions.

Relative to prior structured matrix methods, the new mathematical object is the sign-corrected XOR-slice image of the diagonally modulated shared-profile map in Eq. (1.1). Butterfly and Monarch operators use products of prescribed sparse factors; Kronecker methods use separable tensor factors; circulant and Walsh–Hadamard methods use transform diagonalization; and hypercomplex neural layers tie blocks through multiplication tables [4, 3, 10, 16, 17, 2]. The principal new representation result here is the exact XOR-diagonal compiler together with the fixed-component dimension bound that separates exact representability from parameter-count compression. The cited constructions do not provide this compiler and dimension statement for the present signed-XOR image, nor its joint covariance, differential, and exact-execution analysis. No dominance relation with those families is asserted.

**Contributions.**

1. We formulate representation, initialization, optimization geometry, and exact execution as the image, covariance, differential, and evaluation graph of one shared-profile operator map.
2. We derive ARCANA's unique $1/\sqrt{Rn}$ coefficient scale under explicit Latin-square assumptions and give the corresponding formula for positively homogeneous activations.
3. We connect symmetric component-copy collapse to coherent operator updates, proving the correlation-to-update identity and the conditional $1/R$ first-step learning-rate normalization.
4. For signed XOR, we prove the slice isometry, exact XOR-diagonal compiler, compressive fixed-$R$ measure-zero result, and a strict separation from ordinary low rank.
5. We prove that binary carries restrict fixed stencil offsets to $O(\log n)$ XOR support and retain only minimal numerical checks of the resulting consequences.
6. We state an exact execution contract that distinguishes $W \to W_{\mathrm{AX}}(R)$ approximation from lossless decompilation of $W_{\mathrm{AX}}(R)$ into its execution graph.
7. We prove that the images are nested in component count and formulate a constrained Pareto functional as a possible consequence; no runtime selection policy is validated.

Large-model recovery, diffusion experiments, and platform benchmarking are outside the scope of the paper. No universal optimizer, compression, or hardware-speed claim is made.

## 2 Background and Related Work

**Low-rank approximation and adapters.** For $W \in \mathbb{R}^{m\times n}$, truncated SVD is the optimal rank-$r$ approximation in Frobenius norm [9, 22]. Neural compression and parameter-efficient adaptation exploit coordinate redundancy through pruning, sharing, factorizations, and low-rank updates [11, 13, 6]. Low-rank methods are strongest when singular energy is concentrated; they do not directly exploit a full-rank matrix that is simple in another coordinate system.

**Structured matrix parameterizations.** Displacement structure, circulant operators, block-sparse products, Butterfly, Monarch, tensor-train, and Kronecker constructions trade unrestricted entries for algebraic or compositional sharing [14, 4, 3, 20, 10]. These constructions also show why parameter count, arithmetic cost, and data movement must be stated separately. AXIOM instead uses a fixed signed-XOR routing pattern with diagonal modulation; no dominance over these structured families is claimed.

**Hypercomplex and Clifford networks.** Complex, quaternion, octonion, and more general hypercomplex layers tie real matrix blocks according to multiplication tables [17, 7, 1, 5]. Clifford neural layers similarly use algebra-valued representations in PDE and equivariant settings [2, 21]. AXIOM-TENSOR uses only the coordinate rule in Eq. (1.3): each basis product is a signed XOR shift. The exact compiler relies on signed-permutation support, not on associativity, alternativity, or a claim that nonassociativity benefits learning.

**XOR convolution and transforms.** When $\theta \equiv +1$, Eq. (1.3) becomes convolution on $(\mathbb{Z}_2)^k$, diagonalized by the Walsh–Hadamard transform [16]. The Cayley–Dickson sign twist generally destroys simultaneous diagonalizability: left-basis operators do not commute for $n \geq 4$. This prevents the ordinary $O(n \log n)$ Walsh–Hadamard route, although it is not a lower bound against every conceivable algorithm.

**Initialization and optimization.** Dense Xavier and He rules allocate variance under independent-entry parameterizations [8, 12]. Component-summed sharing changes the number of random reads contributing to an effective entry, while adaptive optimizers act on the redundant coordinates rather than directly on the materialized operator [15]. Sections 5 and 6 derive the corresponding variance and coherent-update laws from the covariance and differential of the same map.

**Compact execution.** Butterfly, Monarch, and Kronecker-sparse operators emphasize that learned-scalar count, working memory, arithmetic, and hardware latency are distinct [4, 3, 10]. A signed-permutation plan can execute reduced AXIOM coefficients without dense materialization, but its straightforward arithmetic remains quadratic. We derive the exact execution theorem and storage crossover while omitting platform benchmarks.

**Scope.** The paper analyzes four views of one operator map and derives nested images as a consequence of varying component count. Results from one view are not used as evidence for another without an explicit argument. The numerical section is limited to checks of selected theoretical consequences; broad model-recovery and hardware studies are outside scope.

# 3 Shared-Profile Operator Map

## 3.1 General component-summed form

Write $[n] = \{0, \ldots, n-1\}$. Fix an index map $\phi : [n] \times [n] \to [m]$ and gains $\vartheta_{ij}$ with $|\vartheta_{ij}| = 1$. A profile $a \in \mathbb{R}^m$ defines

$$S_\phi(a)_{ij} = \vartheta_{ij} a_{\phi(i,j)}. \tag{3.1}$$

The map $S_\phi$ is linear in $a$.

**Definition 3.1** (Component-summed shared-profile operator)**.** For profiles $A_r \in \mathbb{R}^m$ and row and column modulators $U_r, V_r \in \mathbb{R}^n$, define

$$\mathcal{F}_R(A, U, V) = \sum_{r=1}^{R} \operatorname{diag}(U_r) S_\phi(A_r) \operatorname{diag}(V_r). \tag{3.2}$$

When $m = n$, the generic square parameter array has $R(m + 2n) = 3Rn$ learned scalars. The symbol $R$ denotes component count, never ordinary matrix rank.

The image, pushforward moments, differential, and factorized evaluation of this map govern representation, initialization, optimization geometry, and exact execution, respectively. Varying $R$ produces the nested-image family studied in Section 8. Conclusions in one view do not transfer to another without an additional argument.

**Definition 3.2** (Latin-square condition)**.** When $m = n$, the map $\phi$ is Latin-square if $j \mapsto \phi(i, j)$ is a bijection for every row $i$ and $i \mapsto \phi(i, j)$ is a bijection for every column $j$.

Row bijectivity makes distinct entries in a row read distinct profile coordinates; column bijectivity gives the analogous property for the transpose. These assumptions are needed for distribution-robust forward and backward variance statements, not for the definition of the operator.

## 3.2 Signed-XOR specialization

Let $A_1 = \mathbb{R}$ and construct $A_{2q} = A_q \oplus A_q e$ using

$$(a, b)(c, d) = \big(ac - d^* b,\ da + bc^*\big), \qquad (a, b)^* = (a^*, -b). \tag{3.3}$$

The convention fixes $e_0^* = e_0$ and $e_j^* = -e_j$ for $j > 0$. At $n = 2^k$, basis multiplication has the signed-XOR form in Eq. (1.3) [18, 19].

For $a \in \mathbb{R}^n$, left multiplication is

$$(a * x)_i = \sum_{p \oplus q = i} \theta(p, q) a_p x_q \tag{3.4}$$

$$= \sum_{j=0}^{n-1} \theta(i \oplus j, j) a_{i \oplus j} x_j = (L_a x)_i, \tag{3.5}$$

and therefore

$$(L_a)_{ij} = \theta(i \oplus j, j) a_{i \oplus j}. \tag{3.6}$$

This is Eq. (3.1) with

$$\phi(i, j) = i \oplus j, \qquad \vartheta_{ij} = \theta(i \oplus j, j). \tag{3.7}$$

XOR with a fixed index is a permutation, so $\phi$ is Latin-square.

**Definition 3.3** (AXIOM operator)**.** The signed-XOR specialization of Eq. (3.2) is

$$T(a, u, v) = \mathrm{diag}(u) L_a \,\mathrm{diag}(v), \qquad W_{\mathrm{AX}}(R) = \sum_{r=1}^{R} T(A_r, U_r, V_r). \tag{3.8}$$

Its entries are

$$W_{\mathrm{AX}}(R)[i, j] = \sum_{r=1}^{R} U_{ri} \theta(i \oplus j, j) A_{r, i \oplus j} V_{rj}. \tag{3.9}$$

Let $\Sigma_{ij} = \theta(i \oplus j, j)$ and $C_a[i, j] = a_{i \oplus j}$. One component is

$$T(a, u, v) = \Sigma \circ \big(\mathrm{diag}(u) C_a \,\mathrm{diag}(v)\big), \tag{3.10}$$

where $\circ$ is the Hadamard product. Although $\Sigma \circ \Sigma = \mathbf{1}$, entrywise sign removal need not preserve singular values.

### 3.3 Rectangles and the two representation quantifiers

For logical dimensions $d_{\text{in}}, d_{\text{out}}$, choose the next power of two $n$, zero-embed the input, apply the square core, and crop the output. Equivalently, embed the target in the leading rectangle of an $n \times n$ zero matrix. This proves coverage but padding can weaken the parameter budget.

Exact representability asks

$$\forall W \in \mathbb{R}^{n\times n}\ \exists R, A, U, V : W = W_{\text{AX}}(R), \tag{3.11}$$

whereas fixed-budget approximation prescribes $R_0$ and asks for

$$\min_{A,U,V} \frac{\|W - W_{\text{AX}}(R_0)\|_{\text{F}}}{\|W\|_{\text{F}}}. \tag{3.12}$$

The first permits $R$ to grow with $n$; the second prescribes the component count. Neither quantifier implies a runtime advantage.

**Terminology.** An *exact representation* is an equality $W = W_{\text{AX}}(R)$. An *approximation* selects an element of the fixed-$R$ image with nonzero error. We use *compression* only when the stored target-dependent data are smaller than the corresponding dense representation. The *compiler* in Section 4 maps a target matrix to AXIOM coefficients; it may be exact without being compressive. The *decompiler* in Section 7 maps fixed AXIOM coefficients to an execution graph. Its losslessness means exact evaluation of $W_{\text{AX}}(R)$, not exact recovery of the matrix from which those coefficients may have been fitted. In all cases, $R$ denotes component count and $\text{rank}(W)$ denotes ordinary matrix rank.

## 4 Representation Theory and Compression Conditions

### 4.1 The sign-corrected XOR-slice coordinate system

For each offset $p \in \{0, \dots, n-1\}$, define the signed shift

$$S_p[i, j] = \theta(p, j)\mathbf{1}\{i \oplus j = p\}. \tag{4.1}$$

Every $S_p$ is a signed permutation matrix, the supports of the $S_p$ are disjoint, and $L_a = \sum_p a_p S_p$. Thus an ordinary sum of left multiplications has no additional expressive power:

$$\sum_{r=1}^{R} L_{a_r} = L_{\sum_r a_r}. \tag{4.2}$$

The diagonal modulators in Eq. (3.8) are therefore essential.

The relevant coordinates become explicit after a reindexing.

**Definition 4.1** (Sign-corrected XOR slices)**.** For $W \in \mathbb{R}^{n\times n}$, define $X(W) \in \mathbb{R}^{n\times n}$ by

$$X(W)_{p,i} = \theta(p, i \oplus p)W_{i,i\oplus p}. \tag{4.3}$$

The row $X(W)_{p,:}$ is the $p$th sign-corrected XOR diagonal of $W$.

**Proposition 4.2** (XOR-slice isometry and component form)**.** *The map $X$ is a linear Frobenius isometry. Its inverse is*

$$W_{ij} = \theta(i \oplus j, j)X(W)_{i\oplus j,i}. \tag{4.4}$$

*Moreover,*

$$X\big(W_{\text{AX}}(R)\big)_{p,i} = \sum_{r=1}^{R} A_{rp}U_{ri}V_{r,i\oplus p}. \tag{4.5}$$

*Proof.* The map $(p, i) \mapsto (i, j = i \oplus p)$ is a bijection of the $n^2$ matrix coordinates. Multiplication by $\theta(p, j) \in \{\pm 1\}$ preserves squared magnitude, proving the isometry and Eq. (4.4). Substitution of $j = i \oplus p$ into Eq. (3.9) cancels the two equal signs and gives Eq. (4.5). □

Equation (4.5) identifies the relevant sharing structure. In an ordinary rank-$R$ factorization, entries have the form $\sum_r x_{ri} y_{rj}$. In AXIOM coordinates they have the form $\sum_r A_{rp} U_{ri} V_{r,i\oplus p}$. One component may vary along a selected XOR diagonal through $U$, but behavior across multiple diagonals is coupled by the same row and permuted-column profiles. Exact or accurate representation at a reduced component count therefore depends on agreement with this sharing geometry.

## 4.2 Exact programming and its storage meaning

**Lemma 4.3** (Programming one XOR diagonal)**.** *Fix $p$ and arbitrary values $z_0, \ldots, z_{n-1}$. The matrix supported on $(i, i \oplus p)$ with value $z_i$ is one AXIOM component.*

*Proof.* Choose $a = e_p$, $v = \mathbf{1}$, and $u_i = \theta(p, i \oplus p) z_i$. Equation (3.9) is zero away from the $p$th XOR diagonal and equals $z_i$ on it. □

Taking a vector $z$ with one nonzero entry shows that every matrix unit $E_{ij}$ is one component. More usefully, Lemma 4.3 groups all entries having the same XOR offset.

**Theorem 4.4** (Exact XOR-diagonal compiler)**.** *Every $W \in \mathbb{R}^{n\times n}$ has an exact representation with at most $n$ AXIOM components. A deterministic construction uses, for every $p$,*

$$a^{(p)} = e_p, \qquad v^{(p)} = \mathbf{1}, \qquad u_i^{(p)} = \theta(p, i \oplus p) W_{i,i\oplus p}. \tag{4.6}$$

*Then $W = \sum_p T(a^{(p)}, u^{(p)}, v^{(p)})$.*

*Proof.* At $(i, j)$ only $p = i \oplus j$ is active. The two factors $\theta(p, j)$ multiply to one, leaving $W_{ij}$. □

Let $\mathcal{D}(W) = \{p : X(W)_{p,:} \neq 0\}$. Omitting zero slices immediately gives the following support-adaptive form.

**Corollary 4.5** (XOR-support bound)**.** *If $|\mathcal{D}(W)| = m$, then $W$ is represented exactly with at most $m$ components.*

The compiler establishes exact representability but not compression. Its target-dependent vectors $u^{(p)}$ contain exactly $n^2$ values. A specialized representation is therefore storage neutral up to fixed metadata; writing all profiles in the generic trainable format allocates $3n^2$ scalar slots. The sign table is also not the source of exact representability: the proof uses only a known nonzero multiplier on every support location. The Cayley–Dickson signs select the coordinate geometry in which sharing is imposed.

## 4.3 Full rank with one component

**Proposition 4.6** (Generic full matrix rank)**.** *For one component,*

$$\det T(a, u, v) = \Big(\prod_i u_i\Big)\Big(\prod_i v_i\Big) \det L_a. \tag{4.7}$$

*The polynomial* $\det L_a$ *is not identically zero; hence a generic component with nonzero modulators has full ordinary matrix rank.*

*Proof.* Equation (4.7) follows from determinant multiplicativity. At $a = e_0$, left multiplication is the identity, so $\det L_{e_0} = 1$. Therefore the determinant polynomial is nonzero and vanishes only on a proper algebraic subset. □

There is also an explicit separation from low rank.

**Proposition 4.7** (AXIOM–low-rank separation)**.** *For every $p$, $S_p = T(e_p, \mathbf{1}, \mathbf{1})$ is one AXIOM component and has all $n$ singular values equal to one. Its best ordinary rank-$r$ approximation satisfies*

$$\min_{\operatorname{rank}(B)\le r} \frac{\|S_p - B\|_{\mathrm{F}}}{\|S_p\|_{\mathrm{F}}} = \sqrt{1 - \frac{r}{n}}. \tag{4.8}$$

*Proof.* $S_p$ is a signed permutation and therefore orthogonal. Its singular values are all one. The Eckart–Young theorem leaves $n - r$ unit singular values after the optimal rank-$r$ truncation, giving Eq. (4.8). □

Thus small AXIOM component count and small matrix rank measure genuinely different structures rather than two estimators of the same property.

## 4.4 Why fixed-component exact compression is nongeneric

Each component is the image of the rank-one three-way tensor $a \otimes u \otimes v$ under a fixed linear map. This observation yields a global dimension bound, including degenerate parameter choices.

**Proposition 4.8** (Fixed-$R$ image dimension)**.** *The set of matrices representable by $R$ AXIOM components has dimension at most $R(3n - 2)$. If $R(3n - 2) < n^2$, it has empty interior and Lebesgue measure zero in $M_n(\mathbb{R})$.*

*Proof.* The affine cone of rank-one tensors $a \otimes u \otimes v$ has dimension $3n - 2$: three length-$n$ vectors give $3n$ parameters and two independent nonzero rescalings leave their tensor product unchanged. One AXIOM component is a linear image of this cone, so its image dimension is at most $3n - 2$. The addition map from a Cartesian product of $R$ such images has dimension at most $R(3n - 2)$. A semialgebraic subset of smaller dimension than the ambient $n^2$-dimensional space has empty interior and measure zero. □

**Corollary 4.9** (Compression–representation separation)**.** *Whenever the generic triplet count is smaller than dense storage, $3Rn < n^2$, the fixed-$R$ AXIOM image has measure zero. Hence no parameter-count-compressive fixed $R$ exactly represents a generic dense matrix.*

*Proof.* $R(3n - 2) < 3Rn < n^2$, so Proposition 4.8 applies. □

**Corollary 4.10** (Conditional exact structural compression)**.** *If a target already belongs to the $R$-component AXIOM image and $R < n/3$, its generic factors store it exactly with compression ratio*

$$\frac{n^2}{3Rn} = \frac{n}{3R}. \tag{4.9}$$

*This exact compressed family contains full-rank matrices.*

*Proof.* The storage ratio is immediate. Proposition 4.6 shows that already at $R = 1$ a generic component is full rank. □

The measure-zero statement is therefore a target-selectivity result, not an absence-of-compression result. It rules out one reduced family covering every generic dense matrix exactly; it leaves exact structural compression, approximate compression, and native structured training as distinct target-aligned possibilities.

### 4.5 An unstructured negative control

The narrower family $\{L_a : a \in \mathbb{R}^n\}$ is the $n$-dimensional linear span of the orthogonal shifts $S_p$. Because their supports are disjoint and $\|S_p\|_F^2 = n$, the least-squares projection of $W$ is

$$\widehat{a}_p = \frac{1}{n} \sum_{j=0}^{n-1} \theta(p, j) W_{p \oplus j, j}. \tag{4.10}$$

For a matrix with i.i.d. zero-mean entries of common variance, isotropy gives

$$\frac{\mathbb{E}\|L_{\widehat{a}}\|_F^2}{\mathbb{E}\|W\|_F^2} = \frac{1}{n}, \qquad \sqrt{\frac{\mathbb{E}\|W - L_{\widehat{a}}\|_F^2}{\mathbb{E}\|W\|_F^2}} = \sqrt{1 - \frac{1}{n}}. \tag{4.11}$$

The second quantity is a root-mean-square energy ratio, not the expectation of a random ratio. It provides a parameter-free negative-control prediction.

### 4.6 Why local stencils are XOR sparse

Ordinary locality and XOR locality are related by binary carry propagation. For $0 < \delta < n = 2^k$, define

$$D_\delta(n) = \{i \oplus (i + \delta) : 0 \le i < n - \delta\}. \tag{4.12}$$

**Lemma 4.11** (Binary-carry bound)**.** *Let $\ell = \lceil \log_2(\delta + 1) \rceil$. Then*

$$|D_\delta(n)| \le 2^\ell (k - \ell + 1). \tag{4.13}$$

*For fixed $\delta$, this is $O(\log n)$.*

*Proof.* Write $i = 2^\ell h + q$ with $0 \le q < 2^\ell$. Since $\delta < 2^\ell$, adding $\delta$ fixes the low-bit XOR pattern once $q$ is fixed and creates at most one carry into $h$. Without a carry the high-bit XOR is zero. With a carry it is $h \oplus (h + 1) = 2^{t+1} - 1$, where $t$ is the number of trailing ones in $h$. There are at most $k - \ell$ nonzero high patterns, plus the zero pattern. Across the $2^\ell$ possible values of $q$, Eq. (4.13) follows. □

For a power-of-two offset $\delta = 2^q$, the description is exact:

$$i \oplus (i + 2^q) = 2^q \big(h \oplus (h + 1)\big) = 2^q (2^{t+1} - 1), \qquad h = \lfloor i/2^q \rfloor, \tag{4.14}$$

so at most $k - q$ XOR offsets occur. In particular, a tridiagonal stencil has XOR support $\{0, 1, 3, \ldots, 2^k - 1\}$ and therefore at most $k + 1$ exact AXIOM components.

**Corollary 4.12** (Fixed-stencil representation)**.** *Suppose a matrix is supported on a fixed finite set $\Delta$ of ordinary integer offsets, independent of $n$. Then it occupies $O(|\Delta| \log n)$ XOR diagonals and has an exact AXIOM representation with that many components. Power-of-two grid strides obey the sharper count in Eq.* (4.14)*.*

*Proof.* Positive and negative versions of an offset generate the same XOR values after reindexing because XOR is symmetric. Take the union of Lemma 4.11 over $\Delta$ and apply Corollary 4.5. □

**Corollary 4.13** (Exact stencil compression)**.** *Under Corollary 4.12, fixed stencils have an exact generic AXIOM description using $O(n \log n)$ scalars, compared with $n^2$ dense entries. The specialized compiler stores only the $O(n \log n)$ target values on active XOR diagonals and fixed routing metadata. Direct application of its one-offset components costs $O(Bn \log n)$ for batch size $B$.*

*Proof.* There are $m = O(\log n)$ active XOR diagonals. Apply the one-diagonal compiler with $m$ components. Each component stores or applies one length-$n$ value vector and one signed permutation. □

This establishes the mechanism observed for finite-difference matrices: their ordinary rank can be full, yet fixed local offsets create only logarithmically many signed-XOR slices. The conclusion depends on ordering and fixed offsets; it is not a claim about arbitrary sparse matrices.

# 5 ARCANA Initialization as a Pushforward Law

Initialization concerns the distribution induced on operators and activations by random parameters; it does not ask which deterministic matrices belong to the image. We state the result for the general map in Eq. (3.2). The resulting per-component scale is named ARCANA, short for *Algebraic Rank-Corrected Activation-Normalized Amplitude*. "Amplitude" refers to coefficient standard deviation; the activation-normalized extension appears in Theorem 5.2.

## 5.1 Linear forward and backward variance

Assume $m = n$, $U_r = V_r = \mathbf{1}$, and independent profile coefficients $A_{r,p}$ with mean zero and variance $\sigma_A^2$. The coefficients are also independent of the input. At this symmetric point,

$$\mathcal{F}_R(A, \mathbf{1}, \mathbf{1}) = S_\phi(\bar{A}), \qquad \bar{A} = \sum_{r=1}^{R} A_r, \qquad \mathrm{Var}(\bar{A}_p) = R\sigma_A^2. \tag{5.1}$$

**Theorem 5.1** (ARCANA linear-variance law)**.** *Suppose $\phi$ is row bijective and $|\vartheta_{ij}| = 1$. For $y = \mathcal{F}_R x$ and any input second-moment matrix independent of the parameters,*

$$\mathbb{E}[y_i^2] = R\sigma_A^2 \sum_{j=0}^{n-1} \mathbb{E}[x_j^2]. \tag{5.2}$$

*If every input coordinate has variance $\sigma_x^2$, then*

$$\mathrm{Var}(y_i) = nR\sigma_A^2\sigma_x^2. \tag{5.3}$$

*The unique positive coefficient standard deviation preserving this variance is*

$$\boxed{\sigma_A = \frac{1}{\sqrt{Rn}}.} \tag{5.4}$$

*If $\phi$ is also column bijective, the same rule preserves backward coordinate variance for the transpose map.*

*Proof.* For a fixed row $i$, row bijectivity makes distinct columns read distinct coordinates of every random profile. Hence $\mathbb{E}[W_{ij}W_{ij'}] = 0$ for $j \neq j'$, even when $x_j$ and $x_{j'}$ are correlated. Each entry is a sum of $R$ independent profile reads and has second moment $R\sigma_A^2$. Expanding $\mathbb{E}[y_i^2]$ gives Eq. (5.2), and equal coordinate variances give Eq. (5.3). Setting its gain to one has the unique positive solution in Eq. (5.4). Apply the row argument to $\mathcal{F}_R^\top$ for the backward statement. □

The component-count correction is therefore forced by the sum in Eq. (5.1). Using $1/\sqrt{n}$ independently for every component gives output variance $R\sigma_x^2$ and RMS gain $\sqrt{R}$. At $R = 1$, the coefficient variance reduces to the fan-in Xavier value. AXIOM satisfies the Latin-square assumption because XOR by a fixed index is bijective.

### 5.2 Active fan-in and nonlinear moments

If masking or grouping leaves only $q$ active input coordinates per exposed output, replace $n$ by $q$. Forward preservation uses $\sigma_A^2 = 1/(Rq_{\text{in}})$, whereas backward preservation uses $1/(Rq_{\text{out}})$. They coincide only for a balanced wrapper; padding with deterministic zeros must not be counted as active fan-in.

For a nonlinear activation, the moment rather than its name determines the constant.

**Theorem 5.2** (Positively homogeneous activation)**.** *Let $Z \sim \mathcal{N}(0,1)$ and let $\psi$ be positively homogeneous of degree $d > 0$. Put $m_\psi = \mathbb{E}[\psi(Z)^2]$. Under the Gaussian pre-activation approximation and unit input RMS, unit post-activation second moment is obtained by*

$$\boxed{\sigma_A = \frac{m_\psi^{-1/(2d)}}{\sqrt{Rn}}.} \tag{5.5}$$

*Proof.* The pre-activation has standard deviation $s = \sqrt{Rn}\,\sigma_A$. Homogeneity gives $\mathbb{E}[\psi(sZ)^2] = s^{2d} m_\psi$. Equating this expression to one and solving for the coefficient standard deviation gives Eq. (5.5). □

For ReLU, $d = 1$ and $m_\psi = 1/2$, yielding $\sigma_A = \sqrt{2/(Rn)}$. For squared ReLU, $d = 2$ and $m_\psi = 3/2$, yielding

$$\sigma_A = \frac{(2/3)^{1/4}}{\sqrt{Rn}}. \tag{5.6}$$

Writing Eq. (5.5) in terms of coefficient standard deviation is important: its moment exponent is $-1/(2d)$. Random independent modulators with second moments $\mu_U$ and $\mu_V$ multiply the variance budget by $\mu_U \mu_V$.

### 5.3 Boundary of the initialization theorem

The theorem controls marginal second moments at step zero. It does not control singular-value spread, activation tails, residual-depth accumulation, or the Jacobian that maps optimizer steps into matrices. In particular, the same collapse that produces Eq. (5.4) creates coherent gradients, which is the subject of the next section.

## 6 Optimization Geometry from the Differential

Let $\Theta = (A, U, V)$ and consider a small parameter update $\Delta\Theta$. Its first-order function-space effect is

$$\Delta W = D\mathcal{F}_R(\Theta)[\Delta\Theta] + O(\|\Delta\Theta\|^2). \tag{6.1}$$

Initialization controls the distribution of $\mathcal{F}_R(\Theta)$; optimization depends on its differential. These can scale differently with $R$.

### 6.1 Component-copy collapse

**Lemma 6.1** (Symmetric profile-gradient collapse)**.** *At $U_r = V_r = \mathbf{1}$, the forward map depends on the profiles only through $\bar{A} = \sum_r A_r$. For any differentiable loss depending on the operator,*

$$\frac{\partial \mathcal{L}}{\partial A_1} = \cdots = \frac{\partial \mathcal{L}}{\partial A_R}. \tag{6.2}$$

*Proof.* Equation (5.1) follows from linearity of $S_\phi$. Differentiating a function of $\sum_r A_r$ with respect to any summand gives the same derivative. □

If a deterministic optimizer also has identical state for these coordinates, it emits identical profile updates. Their induced operator changes then add in the same direction. Small symmetry-breaking noise, factor-specific gradients, minibatch randomness, and optimizer history can reduce but need not eliminate this coherence.

### 6.2 Correlation-to-update identity

**Theorem 6.2** (Coherent effective update)**.** *Let $z_r$ denote the first-order operator direction contributed by component $r$ and assume*

$$\mathbb{E}\|z_r\|_{\mathrm{F}}^2 = q^2, \qquad \mathbb{E}\langle z_r, z_s\rangle_{\mathrm{F}} = \rho q^2 \quad (r \neq s). \tag{6.3}$$

*For a common step multiplier $\eta$,*

$$\mathbb{E}\|\Delta W\|_{\mathrm{F}}^2 = \eta^2 q^2 R\{1 + (R-1)\rho\}. \tag{6.4}$$

*Proof.* Expand $\|\eta \sum_r z_r\|_{\mathrm{F}}^2$ into $R$ squared-norm terms and $R(R-1)$ ordered cross terms, then substitute the assumed moments. □

For uncorrelated directions ($\rho = 0$), the root-mean-square operator step scales as $\sqrt{R}$. For perfectly coherent directions ($\rho = 1$), it scales as $R$. The identity interpolates between the two and makes the empirical quantity to measure explicit. It is optimizer agnostic: an adaptive optimizer can change $q$ and $\rho$, but summing identical coordinate updates is already coherent under plain gradient descent.

**Corollary 6.3** (Conditional first-step normalization)**.** *In the perfectly coherent regime, if the per-component direction $q$ is otherwise unchanged, matching the first operator step at reference component count $R_0$ and learning rate $\eta_0$ requires*

$$\eta(R) = \eta_0 \frac{R_0}{R}. \tag{6.5}$$

This is a local normalization, not a universal learning-rate schedule. Later updates can decorrelate copies, change curvature, or enter different optimizer states. It also does not follow from ARCANA's $1/\sqrt{R}$ initialization factor: the former normalizes a coherent differential, whereas the latter normalizes the variance of an independent random sum.

### 6.3 Gauges and other redundant directions

Each component admits two continuous rescalings, $(a, u, v) \mapsto (cd\,a, c^{-1}u, d^{-1}v)$. These leave the represented matrix unchanged but alter parameter-space gradients and preconditioning. Permuting components creates additional equivalent parameter points. The operator map can therefore be well scaled at initialization while its parameter metric remains highly anisotropic. Any optimizer claim must be expressed in operator space or state the chosen gauge. The next section fixes the coefficients and considers exact evaluation rather than training dynamics.

## 7 Exact Execution and Storage Contract

The signed shifts $S_p$ from Eq. (4.1) provide a direct arithmetic graph for fixed coefficients at component count $R$. Since $L_{A_r} = \sum_p A_{rp} S_p$,

$$G_R(x) = \sum_{r=1}^{R} \sum_{p=0}^{n-1} A_{rp}\, \mathrm{diag}(U_r) S_p\, \mathrm{diag}(V_r) x. \tag{7.1}$$

We call the coefficient-to-graph construction a *decompiler*. It does not fit a dense target or alter the represented operator.

**Theorem 7.1** (Lossless decompiler correctness)**.** *Under exact arithmetic, the graph in Eq.* (7.1) *satisfies* $G_R(x) = W_{\mathrm{AX}}(R)x$ *for every* $x$*. Rectangular cropping preserves the same identity on the exposed coordinates.*

*Proof.* Each $S_p$ has exactly one signed nonzero per row and column. At matrix coordinate $(i, j)$, only $p = i \oplus j$ contributes, producing precisely the entry in Eq. (3.9). Summing over $r$ completes the equality. □

Thus exact execution and lossless decompilation refer to equality with the selected operator $W_{\mathrm{AX}}(R)$. The word "lossless" applies to the second arrow below, not automatically to the first:

$$W \xrightarrow[\text{generally lossy at prescribed } R]{\text{fit}} W_{\mathrm{AX}}(R) \xrightarrow[\text{exact}]{\text{decompile}} G_R. \tag{7.2}$$

The exact compiler in Theorem 4.4 makes the first arrow exact by allowing up to $n$ specially structured components, but then it retains $n^2$ target values and is not compression.

The generic square plan stores $3Rn$ learned scalars and need not store an $n^2$ sign table because the signs can be generated from the fixed convention. It crosses below $n^2$ dense coefficients only when $R < n/3$. This is precisely the regime in which Corollary 4.9 says generic exact dense coverage is impossible. Compact storage is therefore conditional on having already found a target aligned with the reduced family.

For a dense profile $A_r$, straightforward application costs $\Theta(BRn^2)$ arithmetic for batch size $B$, although it can stream into an $O(Bn)$ accumulator without materializing $W_{\mathrm{AX}}(R)$. Sparse profile support can reduce the work; the one-offset components used for stencils require only a signed permutation and diagonal scaling per component. These statements are an execution contract, not a latency claim. Different floating-point accumulation orders also need not be bit identical even when they evaluate the same exact arithmetic graph.

## 8 Adaptive Representation from Nested Images

Varying the component count produces a nested representation family. This observation gives a mathematical meaning to adaptive representation; it does not, by itself, specify or validate a runtime control system. Let

$$\mathcal{M}_R = \operatorname{Im} \mathcal{F}_R, \qquad E_R(W) = \inf_{Z \in \mathcal{M}_R} \frac{\|W - Z\|_{\mathrm{F}}}{\|W\|_{\mathrm{F}}}. \tag{8.1}$$

Here $W \neq 0$; we set $E_R(0) = 0$ by convention.

**Theorem 8.1** (Nested component-count hierarchy)**.** *For every* $R \geq 1$*,*

$$\mathcal{M}_R \subseteq \mathcal{M}_{R+1}, \qquad E_{R+1}(W) \leq E_R(W). \tag{8.2}$$

*Promotion from* $R$ *to* $R' > R$ *can preserve the represented operator exactly.*

*Proof.* Append a component with zero profile. It contributes the zero matrix, so every $R$-component operator is also an $(R + 1)$-component operator. Taking the infimum over a larger set cannot increase error. Repeating the zero-padding operation gives exact promotion to any $R' > R$. □

This theorem states monotonicity of representable sets and best approximation error. It does not determine which component count should be used in an application. Storage grows as $S_R = R(m+2n)$, equal to $3Rn$ for square AXIOM, while $E_R(W)$ is nonincreasing. The pair therefore defines a discrete accuracy–storage frontier.

If platform-dependent quantities are supplied independently, one possible selection functional over component count and execution form $\rho \in \{\text{dense materialization}, \text{compact graph}\}$, with nonnegative weights, is

$$\min_{R,\rho} \quad \alpha E_R(W) + \beta S(R,\rho) + \gamma L(R,\rho;h) + \delta C(R,\rho) \tag{8.3}$$

subject to quality or resource constraints. Here $S(R,\rho)$ is the storage of the chosen form, $L$ is measured latency on hardware and context $h$, and $C$ is conversion or cache cost. Unlike the nesting relation and coefficient count, $L$ and $C$ are empirical inputs. The functional is a formal consequence of the available representations, not a validated policy.

The representation transitions are asymmetric:

1. dense $W \to W_{\mathrm{AX}}(R)$ is a fit and is generally approximate;
2. AXIOM $\to$ dense materialization and AXIOM $\to G_R$ are exact for the selected operator, up to floating-point accumulation;
3. $R \to R' > R$ is exact by zero-padding, after which new components may be trained;
4. $R \to R' < R$ is a compression step and requires a tolerance test or corrective fine-tuning.

A future runtime implementation could increase the component count when an application-specific error criterion fails and decrease it under a resource constraint. It could also use an exact dense cache on one path and retain the compact execution graph on another. Such switching would require hysteresis near a Pareto boundary; neither the policy nor its thresholds are evaluated in this paper.

Operator equality also does not determine optimizer-state migration. The gauges in Section 6 imply that dense moments cannot in general be inverted into unique AXIOM moments. A runtime transition would have to map moments through a chosen local Jacobian, reinitialize them, or use a correction phase. This choice belongs to an implementation policy, not to the nested-image theorem.

Any empirical evaluation of adaptive representation should report separately the approximation error caused by moving to a smaller-component image and the numerical discrepancy of the graph that evaluates the selected operator. The nesting theorem governs the former; exact execution governs the latter. Neither result validates an end-to-end switching policy.

# 9 Minimal Numerical Checks

These checks are restricted to direct consequences of the signed-XOR representation theory. The ARCANA variance and coherent-update results are proved analytically above and do not rely on an empirical campaign. We do not use large generative models, broad post-hoc recovery, or systems benchmarks.

## 9.1 Algebra and implementation parity

Before measuring approximation, automated tests check the complex, quaternion, and octonion multiplication identities; agreement of recursive and tabulated sign construction through $n = 64$; and agreement of direct, dense-expanded, NumPy, Numba, and Torch implementations. Forward and backward comparisons test that Eq. (3.9) is implemented consistently. These are correctness checks rather than evidence for compression.

## 9.2 Gaussian projection

For each $n \in \{4, 8, 16, 32, 64, 128\}$, we draw 20 matrices with i.i.d. standard Gaussian entries using NumPy seed 0. Each target is projected onto the unmodulated subspace $\{L_a\}$ using Eq. (4.10). We report mean and standard deviation of the realized relative Frobenius error and compare the mean with the root-mean-square scale $\sqrt{1 - 1/n}$ from Eq. (4.11). The theoretical quantity is not treated as the finite-sample expectation of the relative-error ratio.

### 9.3 Stencil mechanism check

We construct 1D Dirichlet and periodic Laplacians, a 2D five-point Laplacian, a 3D seven-point Laplacian, first-order advection matrices, and random banded controls. For each matrix we record its occupied XOR diagonals and fit the diagonally modulated family by alternating least squares over $R \in \{1, 2, \ldots, 14, 16, 18, 20, 24\}$, with two restarts and 90 iterations. "Observed exact" means relative Frobenius error below $10^{-6}$; it is an optimization threshold, whereas Corollary 4.5 gives the exact algebraic upper bound without fitting. The low-rank baseline uses truncated SVD with $r = \mathrm{round}(3R/2)$, matching $2nr$ low-rank parameters to approximately $3nR$ AXIOM parameters. A scaling sweep of the 1D Dirichlet Laplacian uses $n = 16, \ldots, 512$.

### 9.4 Small train-from-scratch compatibility check

The only downstream experiment replaces the middle $256 \times 256$ layer of an MNIST multilayer perceptron. The alternatives are a dense affine layer, a rank-16 factorization, and an 11-component AXIOM layer. The network has the form $784 \to 256 \to 256 \to 10$ with ReLU activations. It is trained end-to-end on a fixed 20,000-example subset for four epochs using Adam with learning rate $10^{-3}$ and batch size 256, then evaluated on the 10,000-example test set. We report CPU mean and sample standard deviation over seeds 1–5. The structured layers have similar budgets (8,448 and 8,704 parameters), while the dense middle layer has 65,792. This experiment asks only whether learning can adapt to the parameterization on a small task.

## 10 Results

### 10.1 Compression under stated target conditions

The generic limitation in Corollary 4.9 does not describe the aligned or task-tolerant cases. Table 1 collects one representative result from each of three distinct regimes. Their acceptance criteria are not interchangeable. The 1D stencil has an exact full-matrix-rank representation with a conservative 9.5× generic-triplet storage ratio. At a 1.33× budget reduction, the two-sided AXIOM fit to a $128 \times 128$ Gaussian matrix has error 0.204 versus 0.459 for parameter-matched low rank. Native MNIST training uses 7.6× fewer middle-layer parameters than the dense model; the reported accuracy intervals overlap, so no ordering is inferred.

Table 1: Minimal checks of theoretically distinct compression regimes. The rows use different acceptance criteria: exact matrix equality, approximation error, and downstream accuracy. They should not be collapsed into one metric.

| Regime | Target / budget | Reduction | Retained criterion |
|---|---|---|---|
| Exact structure | 1D stencil, $n = 256$, $m = 9$ | 9.5× | Exact by XOR-support compiler; full matrix rank. |
| Matched approximation | Gaussian $128^2$, 32 components | 1.33× | Error 0.204 versus 0.459 for parameter-matched low rank. |
| Native training | MNIST $256^2$, 11 components | 7.6× | $0.9600 \pm 0.0027$ accuracy; dense is $0.9582 \pm 0.0041$. |

### 10.2 The unstructured control follows the predicted scale

The mean Gaussian projection error rises from 0.860 at $n = 4$ to 0.969 at $n = 16$, 0.993 at $n = 64$, and 0.996 at $n = 128$. The corresponding $\sqrt{1 - 1/n}$ values are 0.866, 0.968, 0.992, and 0.996. Figure 1(a) shows the full sweep. Thus the $n$-dimensional unmodulated left-multiplication subspace captures only a vanishing fraction of isotropic matrix energy, as predicted. This result is a negative control for $L_a$ and is not an error formula for the modulated multi-component family.

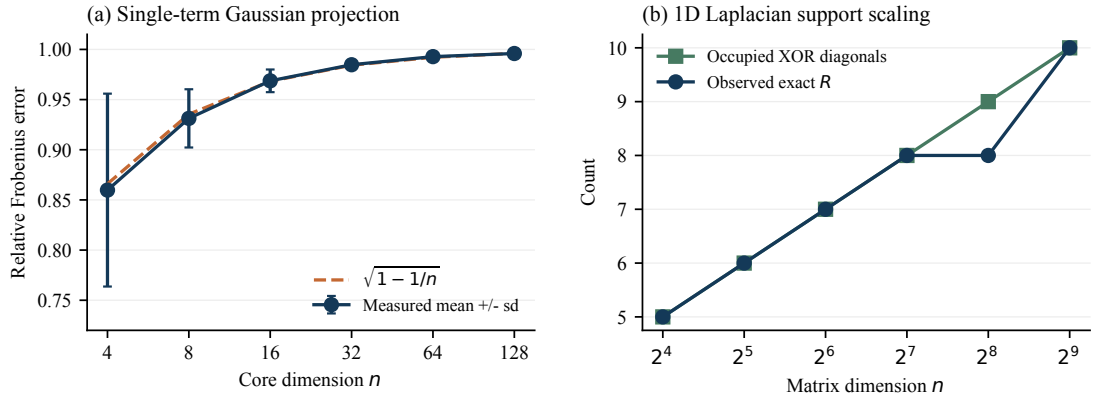



Figure 1: Minimal theory checks. (a) Projection of Gaussian matrices onto the unmodulated left-multiplication subspace follows the predicted $\sqrt{1-1/n}$ scale. Error bars show one standard deviation over 20 draws. (b) For the 1D Dirichlet Laplacian, occupied XOR support equals $\log_2 n + 1$ and the observed numerical exactness threshold tracks that count.

### 10.3 Binary carry explains the stencil results

The 1D Dirichlet Laplacian has ordinary offsets 0 and ±1. By Eq. (4.14), its XOR support is exactly $\{0, 1, 3, \ldots, 2^k - 1\}$, so the observed count is $k + 1$. Across $n = 16, 32, 64, 128, 256, 512$, the recorded support counts are respectively $5, 6, 7, 8, 9, 10$; the smallest fitted component counts reaching $10^{-6}$ are $5, 6, 7, 8, 8, 10$. The single lower count at $n = 256$ reflects numerical approximation just below the declared threshold, not a contradiction of the support upper bound.

The same mechanism persists in higher dimensions. The 2D five-point operator at $n = 256$ occupies nine XOR diagonals, and the 3D seven-point operator at $n = 512$ occupies ten. Both are recovered below $10^{-6}$ with at most nine fitted components. A random nonsymmetric bandwidth-one control also occupies nine XOR diagonals and reaches numerical precision at $R = 9$, showing that constant coefficients and symmetry are not required for the constructive support result. The matched low-rank errors in Table 2 remain large because these matrices have broad singular spectra.

Table 2: Selected full-rank stencil and banded operators. "Fitted components" is the smallest component count with relative error below $10^{-6}$; XOR support is the constructive upper bound. The truncated-SVD rank is chosen to approximately match parameter counts.

| Operator | $n$ | # XOR | fitted components | matched LR error |
|---|---|---|---|---|
| 1D Laplacian, Dirichlet | 256 | 9 | 8 | 0.936 |
| 2D five-point Laplacian | 256 | 9 | 8 | 0.929 |
| 3D seven-point Laplacian | 512 | 10 | 9 | 0.960 |
| central advection | 256 | 8 | 8 | 0.952 |
| random banded, $b = 1$, nonsym. | 256 | 9 | 9 | 0.886 |
| random banded, $b = 2$, sym. | 256 | 16 | 16 | 0.808 |

### 10.4 Small neural compatibility check

The 11-component AXIOM middle layer attains test accuracy $0.9600 \pm 0.0027$ with 8,704 parameters. The dense middle layer attains $0.9582 \pm 0.0041$ with 65,792 parameters, while the matched rank-16 layer attains $0.9573 \pm 0.0046$ with 8,448 parameters. The observed intervals overlap. The result shows only that this small network can adapt to an AXIOM-constrained layer at a reduced parameter count; it does not establish an ordering relative to low rank or behavior on larger tasks.

Table 3: Training from scratch on MNIST: the middle $256 \times 256$ layer is replaced by each variant and the network is trained end-to-end. Test accuracy is mean±sample standard deviation over five CPU seeds. The intervals overlap; this compatibility check does not establish an ordering.

| Middle layer | Parameters | Test accuracy (5 seeds) |
|---|---|---|
| Dense | 65,792 | $0.9582 \pm 0.0041$ |
| Low-rank ($k$=16) | 8,448 | $0.9573 \pm 0.0046$ |
| AXIOM (11 components) | 8,704 | $0.9600 \pm 0.0027$ |

## 11 Discussion

**One map, four analytic views, and one nested-image consequence.** The unified formulation does not assert that representation, initialization, optimization, and execution are the same problem. It identifies the common object from which their different questions arise. The image of $\mathcal{F}_R$ determines attainable matrices; its pushforward determines initial moments; its differential determines local function-space motion; and its factorization determines an arithmetic graph. Varying component count produces nested images. Conclusions do not cross between these views without an additional argument.

**The same collapse explains initialization and update scaling.** At equal diagonal gains, $R$ profiles enter the forward operator only through their sum. Independent random profiles therefore add in variance, forcing the $1/\sqrt{R}$ initialization factor. The derivatives with respect to those profiles are equal, so synchronized updates add in norm and can approach an $R$ factor. There is no contradiction: one statement concerns an independent random sum at step zero, the other a correlated update sum after differentiation. A layer can have component-count-invariant activation RMS and component-count-dependent learning dynamics simultaneously.

**Signed-XOR slices supply the representation mechanism.** The useful coordinate is not ordinary matrix rank but $X(W)_{p,i}$. One component contributes $A_{rp}U_{ri}V_{r,i\oplus p}$: it can carry a full vector on a chosen XOR diagonal, but many diagonals must share the same row and permuted-column profiles. This explains both the exact compiler and the fixed-budget geometry. Reduced storage is available when the target reuses these degrees of freedom across offsets.

**AXIOM and low rank encode different simplicity.** A signed XOR shift costs one AXIOM component, is full rank, and has flat singular spectrum. Conversely, a low-rank dense matrix need not have sparse or shared XOR slices. Neither low-budget family uniformly contains the other. The $128 \times 128$ matched-budget Gaussian fit, where AXIOM error is 0.204 and low-rank error is 0.459, is a concrete example that the dimension theorem does not impose an approximation ordering.

**Binary carries make locality visible in XOR coordinates.** Adding a fixed integer offset changes a bounded low-bit pattern and a run of higher carry bits, producing only logarithmically many XOR differences. PDE semantics, symmetry, and constant coefficients are not the cause; fixed offsets and ordering are. Fixed stencils consequently achieve exact $O(n \log n)$ storage and application despite full ordinary matrix rank.

**Nested images formalize adaptive representation.** The fixed-$R$ measure-zero theorem is not a verdict that AXIOM cannot compress. It states that reduced-component exact representation is target selective. Nested images show that increasing $R$ cannot worsen the best approximation, whereas

decreasing $R$ reduces coefficient storage and requires a stated error criterion. Exact promotion by zero-padding preserves the selected operator. These facts define a representation hierarchy; they do not validate a runtime selection policy. Table 1 separately records exact structural, approximate, and native-training reductions.

**Exact execution begins after the approximation decision.** Once $(A, U, V)$ are fixed, the signed-permutation graph evaluates their operator without additional modeling error. That fact does not repair a lossy dense fit, but it preserves the selected structured representation. Avoiding dense materialization also does not guarantee lower latency: storage, approximation, arithmetic, and measured hardware time are separate quantities.

**Role of the retained evidence.** Gaussian projection and stencils check target-class predictions. MNIST checks compatibility with training the constrained layer on one small task. Exact matrix equality, approximation error, and downstream accuracy are different acceptance criteria and are not aggregated into a single empirical claim.

## 12 Scope Conditions and Open Problems

The following conditions delimit which compression mode is being claimed; they do not negate the exact structural, approximate, or native-training reductions in Table 1.

1. **The unrestricted exact compiler is storage neutral.** Its specialized form retains $n^2$ target-dependent values. Compression instead comes from smaller XOR support, a target at reduced component count, or an accepted approximation criterion.

2. **The dimension bound is not an approximation bound.** Measure zero rules out generic exact recovery at compressive $R$ but does not give the distance of a particular matrix to the fixed-$R$ image or compare it with low rank.

3. **ARCANA is a moment theorem.** It assumes independent, zero-mean profile coefficients and the stated bijectivity conditions. The nonlinear constants additionally use a Gaussian pre-activation approximation. Unit variance does not imply dynamical isometry or convergence.

4. **The optimization law is local and conditional.** The correlation identity is exact under its moment assumptions, but the $1/R$ learning-rate rule requires near-perfect coherence and unchanged per-component direction. Later optimizer states can change that scaling.

5. **The stencil fit is nonconvex.** "Observed exact" uses a $10^{-6}$ numerical threshold. The algebraic XOR-support compiler and its exact compression bound are independent of that optimizer.

6. **The carry theorem depends on indexing.** Reordering degrees of freedom, irregular meshes, or offsets growing with $n$ can enlarge XOR support. Ordinary sparsity alone is insufficient.

7. **The neural evidence is selective.** MNIST is small and does not establish broad capability preservation or post-hoc compressibility of a pretrained model.

8. **Exact execution is not a speed theorem.** A dense-profile graph remains $\Theta(BRn^2)$, although sparse XOR profiles can be subquadratic.

9. **The nested hierarchy is not a validated runtime controller.** Component-count monotonicity and exact promotion are theorems. Policy weights, hysteresis, optimizer-state migration, and hardware-aware routing remain unvalidated and require end-to-end evaluation.

# 13 Conclusion

The paper proves representation, initialization, optimization-geometric, and execution results for one component-summed shared-profile map. In the signed-XOR specialization, the XOR-slice transform is a Frobenius isometry, an arbitrary XOR diagonal is programmable by one component, and every dense matrix has an exact $n$-component compiler. The compiler is not generally a compression: every parameter-count-compressive fixed-$R$ image has measure zero in matrix space. A single component can nevertheless have full matrix rank, and fixed ordinary stencil offsets occupy only $O(\log n)$ XOR diagonals. The same operator map yields the ARCANA variance scale, the coherent-update identity for its differential, and an exact execution graph for fixed coefficients. Exact execution concerns the selected AXIOM operator, not the accuracy of a preceding dense-to-AXIOM fit. Finally, the images are nested in component count, which gives monotone best-approximation error and exact promotion by zero-padding.

The numerical evidence is limited to checks of these statements and their implementation. Gaussian projection onto the unmodulated subspace follows the predicted energy scale. The observed component counts for stencil matrices track XOR support at the stated $10^{-6}$ threshold, while matched low-rank approximations remain inaccurate for the reported full-rank examples. The five-seed MNIST experiment shows compatibility with training one constrained layer on a small task; overlapping intervals do not establish an ordering among the dense, low-rank, and AXIOM alternatives.

Open problems include quantitative approximation bounds inside the fixed-$R$ image, optimization behavior beyond the local coherent-update regime, optimizer-state migration between representations, and execution methods that exploit sparse XOR profiles. A runtime rule for selecting component count or execution form would also require independent latency, conversion-cost, and quality measurements. The nested-image theorem supplies the representation ordering for such a study but does not constitute a validated controller.

# A Additional Derivations

## A.1 Coordinate check for the exact compiler

For Eq. (4.6), the $p$th component is

$$T_p[i,j] = u_i^{(p)}\theta(i\oplus j, j)a_{i\oplus j}^{(p)}v_j^{(p)} \tag{A.1}$$

$$= \theta(p, i\oplus p)W_{i,i\oplus p}\theta(p,j)\mathbf{1}\{i\oplus j = p\}. \tag{A.2}$$

On the support, $j = i \oplus p$ and the signs square to one. Off the support, the basis profile vanishes. Summing over $p$ visits every matrix coordinate exactly once. For an arbitrary known nonzero multiplier system, its reciprocal replaces the second sign correction.

## A.2 The linear map from rank-one three-way tensors

Define a fixed linear map $\Phi : \mathbb{R}^{n\times n\times n} \to \mathbb{R}^{n\times n}$ on elementary tensors by

$$\Phi(Z)_{ij} = \theta(i\oplus j, j)Z_{i\oplus j,i,j}. \tag{A.3}$$

Then $T(a,u,v) = \Phi(a\otimes u\otimes v)$. This makes the dimension argument in Proposition 4.8 independent of a particular normalization of $a, u, v$. Degenerate triples are already included in the rank-one tensor cone, whose projectivization is the Segre variety $\mathbb{P}^{n-1}\times\mathbb{P}^{n-1}\times\mathbb{P}^{n-1}$.

## A.3 Rectangles and affine maps

For $W \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$, choose the next power-of-two $n$, embed $W$ into the leading rectangle of an $n\times n$ zero matrix, compile it, zero-pad the input, and crop the output. Bias remains an ordinary affine vector. This proves rectangular coverage but does not imply that padding is economical.

### A.4 Why forward equivalence does not imply optimizer equivalence

If a compiled module materializes the same matrix $W$ and bias $b$, forward values and input gradients agree because $\partial \ell / \partial x = W^\top (\partial \ell / \partial y)$. Optimizing the redundant factors directly is different: the factor-to-matrix Jacobian changes the metric and contains scaling directions. Exact function representation therefore does not imply identical training trajectories.

## B Consistency Audit of the Unified Views

The five analytic views use different properties of the same frozen shared-profile object. The unified notation makes their assumptions explicit:

| **Analysis** | Required structure | Unified object |
|---|---|---|
| **ARCANA initialization** | unit-modulus gains, independent profiles; row/column bijection for robust forward/backward laws | covariance of $\mathcal{F}_R(A, \mathbf{1}, \mathbf{1})$ |
| **Coherent-update optimization** | linear profile map, exchangeable copies, symmetric gains for exact collapse | differential $D\mathcal{F}_R$ |
| **AXIOM representation** | signed-XOR map and diagonal modulation | image $\operatorname{Im} \mathcal{F}_R$ and slice transform $X$ |
| **Exact execution** | fixed valid AXIOM coefficients and the same sign convention | arithmetic graph $G_R$ |
| **Adaptive representation** | nested component-count images; any policy requires separate measured inputs | family $\{\mathcal{M}_R\}_R$ and optional selection functional over $(R, \rho)$ |

Three terminology rules remove apparent contradictions. First, $R$ is always component count, not matrix rank. Second, "exact compilation" may refer to the storage-neutral $n$-component map from arbitrary $W$, whereas "lossless decompilation" refers to exact evaluation of an already selected $W_{\mathrm{AX}}(R)$; Eq. (7.2) separates them. Third, ARCANA's $1/\sqrt{R}$ factor normalizes an independent random sum, while inverse-$R$ learning-rate scaling normalizes a perfectly coherent local update. They act on different statistical objects.

For homogeneous activations, consistent use of coefficient standard deviation gives Eq. (5.5) with exponent $-1/(2d)$. It is the unique general expression reproducing both the ReLU and squared-ReLU special cases.

## References

[1] John C. Baez. On quaternions and octonions: Their geometry, arithmetic, and symmetry by john h. conway and derek a. smith. *Bulletin of the American Mathematical Society*, 42(2):229–243, 2005.

[2] Johannes Brandstetter, Rianne van den Berg, Max Welling, and Jayesh K. Gupta. Clifford neural layers for pde modeling. In *International Conference on Learning Representations*, 2023.

[3] Tri Dao, Beidi Chen, Nimit S. Sohoni, Arjun Desai, Michael Poli, Jessica Grogan, Alexander Liu, Aniruddh Rao, Atri Rudra, and Christopher Ré. Monarch: Expressive structured matrices for efficient and accurate training. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 4690–4721. PMLR, 2022.

[4] Tri Dao, Albert Gu, Matthew Eichhorn, Atri Rudra, and Christopher Ré. Learning fast algorithms for linear transforms using butterfly factorizations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1517–1527. PMLR, 2019.

[5] Leonard E. Dickson. On quaternions and their generalization and the history of the eight square theorem. *Annals of Mathematics*, 20(3):155–171, 1919.

[6] Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4133–4145. Association for Computational Linguistics, 2023.

[7] David García-Retuerta, Roberto Casado-Vara, Angel Martin-del Rey, Fernando De La Prieta, Javier Prieto, and Juan M. Corchado. Quaternion Neural Networks: State-of-the-Art and Research Challenges. In Cesar Analide, Paulo Novais, David Camacho, and Hujun Yin, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2020*, volume 12490, pages 456–467. Springer International Publishing, Cham, 2020.

[8] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256. PMLR, 2010.

[9] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4 edition, 2013.

[10] Antoine Gonon, Léon Zheng, Pascal Carrivain, and Tung Quoc Le. Fast inference with kronecker-sparse matrices. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 20075–20102. PMLR, 2025.

[11] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A. Horowitz, and William J. Dally. EIE: Efficient inference engine on compressed deep neural network. *ACM SIGARCH Computer Architecture News*, 44(3):243–254, October 2016.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[13] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[14] Thomas Kailath and Ali H. Sayed. Displacement Structure: Theory and Applications. *SIAM Review*, 37(3):297–386, September 1995.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[16] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The Theory of Error-Correcting Codes*, volume 16. Elsevier, 1977.

[17] Titouan Parcollet, Mirco Ravanelli, Mohamed Morchid, Georges Linarès, Chiheb Trabelsi, Renato De Mori, and Yoshua Bengio. Quaternion recurrent neural networks. In *International Conference on Learning Representations*, 2019.

[18] Park Junhu. AXIOM: Algebraic eXtension by Index–Ordered Multiplication. Master's thesis, Seoul National University, 2026.

[19] Park Junhu. Signed-XOR Structural Constants and Table-Free Multiplication in Clifford and Cayley-Dickson Algebras. SNU Open Repository, `https://s-space.snu.ac.kr/handle/10371/236376`, June 2026. Preprint.

[20] Shikai Qiu, Andres Potapczynski, Marc Anton Finzi, Micah Goldblum, and Andrew Gordon Wilson. Compute better spent: Replacing dense layers with structured matrices. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 41698–41716. PMLR, 2024.

[21] David Ruhe, Johannes Brandstetter, and Patrick Forré. Clifford group equivariant neural networks. *Advances in Neural Information Processing Systems*, 36:62922–62990, 2023.

[22] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra, Twenty-fifth Anniversary Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, January 2022.