Appendix: Symbols and Glossary

Daniel Ari Friedman
ORCID: 0000-0001-6232-9096
Email: daniel@activeinference.institute

August 14, 2025

Contents

1 Appendix: Symbols and Glossary

This appendix consolidates the symbols, variables, and constants used throughout the manuscript.

1.1 Sets and Spaces

Symbol	Name
$\mathbb{R}^n$	Euclidean space
IVM	Isotropic Vector Matrix
Coxeter.4D	Euclidean 4D (E^4)
Einstein.4D	Minkowski spacetime (3+1)
Fuller.4D	Synergetics/Quadray tetrahedral space

Descriptions:

- $\mathbb{R}^n$: n-dimensional real vector space.
- IVM: Quadray integer lattice (CCP sphere centers).
- Coxeter.4D: Four-dimensional Euclidean geometry (not spacetime); see Coxeter, Regular Polytopes (Dover ed., p. 119); related lattice/packing background in Conway & Sloane.
- Einstein.4D: Relativistic spacetime with Minkowski metric.
- Fuller.4D: Quadrays with projective normalization and IVM unit conventions.

1.2 Quadray Coordinates and Geometry

Symbol	Name	Description
$q = (a, b, c, d)$	Quadray point	Non-negative coordinates with at least one zero after normalization
A, B, C, D	Quadray axes	Canonical tetrahedral axes mapped by the embedding
k	Normalization offset	$k = \min(a, b, c, d)$ used to set $q' = q - (k, k, k, k)$
q'	Normalized Quadray	Canonical representative with at least one zero and non-negative entries
$P_0, \ldots, P_3$	Tetrahedron vertices	Vertices used in volume formulas
d_{ij}	Pairwise distances	Distance between vertices P_i and P_j (squared in CM matrix)
$\det(\cdot)$	Determinant	Determinant of a matrix
$\lvert\cdot\rvert$	Magnitude	Absolute value (determinant magnitude)
V_{ivm}	Tetravolume (IVM)	Tetrahedron volume in synergetics/IVM units; unit regular tetra has $V_{ivm} = 1$
V_{xyz}	Tetravolume (XYZ)	Euclidean tetrahedron volume
$S3$	Scale factor	$S3 = \sqrt{9/8}$ with $V_{ivm} = S3\, V_{xyz}$ (synergetics unit convention)
Coxeter.4D	Namespace	Euclidean E^4; regular polytopes
Einstein.4D	Namespace	Minkowski spacetime (metric analogy only here)
Fuller.4D	Namespace	Quadrays/IVM; integer tetravolume
Eq. (lattice_det)	Lattice determinant	Integer-lattice volume via 3x3 determinant
Eq. (ace5x5)	Tom Ace 5x5	Direct IVM tetravolume from Quadrays
Eq. (cayley_menger)	Cayley–Menger	Length-based formula: 288 V^2 = det(·)

1.3 Optimization and Algorithms

Symbol	Name
α	Reflection coefficient
γ	Expansion coefficient
ρ	Contraction coefficient
σ	Shrink coefficient
V_{ivm}	Integer volume monitor

Descriptions:

- $\alpha, \gamma, \rho, \sigma$: Nelder–Mead parameters (typical values 1, 2, 0.5, 0.5).
- V_{ivm}: Tracks simplex volume across iterations.

1.4 Information Theory and Geometry

Symbol	Name	Description
log	Natural logarithm	Logarithm base e
$\mathbb{E}[\cdot]$	Expectation	Mean with respect to a distribution
$F_{i,j}$	Fisher Information entry	Empirical/expected $\mathbb{E}[\partial_{\theta_i} \log p \, \partial_{\theta_j} \log p]$; Eq. (**??**)
$\mathcal{F}$	Variational free energy	$-\log P(o \mid s) + \mathrm{KL}[Q(s) \,\|\, P(s)]$; Eq. (**??**)
$\mathrm{KL}[Q \,\|\, P]$	Kullback–Leibler divergence	$\sum Q \log(Q/P)$; information distance
$\nabla_\theta L$	Gradient	Gradient of loss L with respect to parameters θ (column vector)
η	Step size	Learning-rate scalar used in updates
θ	Parameters	Model parameter vector; indices θ_i
ds^2	Minkowski line element	$-c^2\,dt^2 + dx^2 + dy^2 + dz^2$; Eq. (**??**)
c	Speed of light	Physical constant appearing in Minkowski metric

1.5 Embeddings and Distances

Symbol	Name	Description
M	Embedding matrix	Linear map from Quadray to $\mathbb{R}^3$ (Urner-style unless noted)
$\|\cdot\|_2$	Euclidean norm	$\sqrt{x_1^2 + \cdots + x_n^2}$
R, D	Edge scales	Cube edge R and Quadray edge D with $D = 2R$ (common convention)

1.6 Greek Letters (usage)

Symbol	Name	Description
$\alpha, \gamma, \rho, \sigma$	NM coefficients	Nelder–Mead parameters (reflection, expansion, contraction, shrink)
θ	Theta	Parameter vector in models and metrics
μ	Mu	Internal states (Active Inference)
ψ	Psi	External states (Active Inference)
η	Eta	Step size / learning rate

1.7 Notes (usage and cross-references)

- **Figures referenced**: In-text carry identifiers (e.g., Figure **??**).
- **Equation references**: Use labels defined in the text (e.g., Eq. (**??**)).
- **Namespaces**: We use Coxeter.4D, Einstein.4D, Fuller.4D consistently to designate Euclidean E^4, Minkowski spacetime, and Quadray/IVM synergetics, respectively. This avoids conflation of Euclidean 4D objects (e.g., tesseracts) with spacetime constructs and synergetic tetravolume conventions.

- **External validation**: Cross-reference implementations from the 4dsolutions ecosystem including `qrays.py`, `tetravolume.py`, and educational notebooks in School_of_Tomorrow.
- **Multi-language implementations**: Rust (rusty_rays), Clojure (synmods), POV-Ray (quadcraft.py), and VPython (BookCovers) provide algorithmic verification and performance comparison baselines.

1.8 Acronyms and abbreviations

Acronym	Meaning
CM	Cayley–Menger (determinant-based tetrahedron volume)
PdF	Piero della Francesca (Heron-like tetrahedron volume)
GdJ	Gerald de Jong (Quadray-native tetravolume expression)
K-FAC	Kronecker-Factored Approximate Curvature (optimizer using structured Fisher)
CCP	Cubic Close Packing (same centers as FCC)
FCC	Face-Centered Cubic (same centers as CCP)
E^4	Four-dimensional Euclidean space (Coxeter.4D)
NM	Nelder–Mead (simplex optimization algorithm)
4dsolutions	Kirby Urner's GitHub organization with extensive Quadray implementations
BEAST	Synergetic modules (B, E, A, S, T) in Fuller's hierarchical system
OCN	Oregon Curriculum Network (educational framework integrating Quadrays)
POV-Ray	Persistence of Vision Raytracer (used in quadcraft.py visualizations)

1.9 API Index (auto-generated; Methods linkage)

The table below enumerates public symbols from `src/` modules.

Module	Symbol	Kind	Signature	Summary
`cayley_menger`	`ivm_tetra_volume_cayley_menger`	function	`(d2)`	Compute IVM tetravolume from squared distances via Cayley–Menger.
`cayley_menger`	`tetra_volume_cayley_menger`	function	`(d2)`	Compute Euclidean tetrahedron volume from squared distances (Coxeter.4D).
`conversions`	`quadray_to_xyz`	function	`(q, M)`	Map a `Quadray` to Cartesian XYZ via a 3x4 embedding matrix (Fuller.4D -> Coxeter.4D slice).

Module	Symbol	Kind	Signature	Summary
conversions	urner_embedding	function	(scale)	Return a 3x4 Urner-style symmetric embedding matrix (Fuller.4D -> Coxeter.4D slice).
discrete_variational	DiscretePath	class	| Optimization trajectory on the integer quadray lattice. | | `discrete_variational` | `OptionalMoves` | class |	
discrete_variational	apply_move	function	(q, delta)	Apply a lattice move and normalize to the canonical representative.
discrete_variational	discrete_ivm_descent	function	(objective, start, moves=, max_iter=, on_step=)	Greedy discrete descent over the quadray integer lattice.
discrete_variational	neighbor_moves_ivm	function	()	Return the 12 canonical IVM neighbor moves as Quadray deltas.
examples	example_cuboctahedron_neighbors	function	()	Return twelve-around-one IVM neighbors (vector equilibrium shell).
examples	example_cuboctahedron_vertices_xyz	function	()	Return XYZ coordinates for the twelve-around-one neighbors.
examples	example_ivm_neighbors	function	()	Return the 12 nearest IVM neighbors as permutations of {2,1,1,0} (Fuller.4D).
examples	example_optimize	function	()	Run Nelder–Mead over integer quadrays for a simple convex objective (Fuller.4D).
examples	example_partition_tetravolume	function	(mu, s, a, psi)	Construct a tetrahedron from the four-fold partition and return tetravolume (Fuller.4D).

Module	Symbol	Kind	Signature	Summary
`examples`	`example_volume`	function	`()`	Compute the unit IVM tetrahedron volume from simple quadray vertices (Fuller.4D).
`geometry`	`minkowski_interval`	function	`(dt, dx, dy, dz, c)`	Return the Minkowski interval squared ds^2 (Einstein.4D).

Module	Symbol	Kind	Signature	Summary
`glossary_gen`	`ApiEntry`	class		
			`glossary_gen`	`build_api_index`	function	`(src_dir)`	
			`glossary_gen`	`generate_markdown_table`	function	`(entries)`	
			`glossary_gen`	`inject_between_markers`	function	`(markdown_text, begin, end, payload)`		
			`information`	`action_update`	function	`(action, free_energy_fn, step_size, epsilon)`	Continuous-time action update: da/dt = - dF/da.
			`information`	`finite_difference_gradient`	function	`(function, x, epsilon)`	Compute numerical gradient of a scalar function via central differences.
			`information`	`fisher_information_matrix`	function	`(gradients)`	Estimate the Fisher information matrix via sample gradients.
			`information`	`free_energy`	function	`(log_p_o_given_s, q, p)`	Variational free energy for discrete latent states.
			`information`	`natural_gradient_step`	function

Module	Symbol	Kind	Signature	Summary
nelder_mead_quadray	centroid_excluding	function	(vertices, exclude_idx)	Integer centroid of three vertices, excluding the specified index.
nelder_mead_quadray	compute_volume	function	(vertices)	Integer IVM tetra-volume from the first four vertices.
nelder_mead_quadray	nelder_mead_quadray	function	(f, initial_vertices, alpha, gamma, rho, sigma, max_iter, tol, on_step)	Nelder–Mead on the integer quadray lattice.
nelder_mead_quadray	order_simplex	function	(vertices, f)	Sort vertices by objective value ascending and return paired lists.
nelder_mead_quadray	project_to_lattice	function	(q)	Project a quadray to the canonical lattice representative via normalize.
paths	get_data_dir	function	()	Return quad-math/output/data path and ensure it exists.
paths	get_figure_dir	function	()	Return quad-math/output/figures path and ensure it exists.
paths	get_output_dir	function	()	Return quadmath/output path at the repo root and ensure it exists.
paths	get_repo_root	function	(start)	Heuristically find repository root by walking up from start.
quadray	DEFAULT_EMBEDDING	constant	| | | `quadray` | `Quadray` | class |	Quadray vector with non-negative components and at least one zero (Fuller.4D).
quadray	ace_tetravolume_5x5	function	(p0, p1, p2, p3)	Tom Ace 5x5 determinant in IVM units (Fuller.4D).
quadray	dot	function	(q1, q2, embedding)	Return Euclidean dot product <q1,q2> under the given embedding.

Module	Symbol	Kind	Signature	Summary
quadray	integer_tetra_volume	function	(p0, p1, p2, p3)	Compute integer tetra-volume using det[p1-p0, p2-p0, p3-p0] (Fuller.4D).
quadray	magnitude	function	(q, embedding)	Return Euclidean magnitude ||q|| under the given embedding (vector norm).
quadray	to_xyz	function	(q, embedding)	Map quadray to R^3 via a 3x4 embedding matrix (Fuller.4D -> Coxeter.4D slice).
symbolic	cayley_menger_volume_symbolic	function	(d2)	Return symbolic Euclidean tetrahedron volume from squared distances.
symbolic	convert_xyz_volume_to_ivm_symbolic	function	(V_xyz)	Convert a symbolic Euclidean volume to IVM tetravolume via S3.
visualize	animate_discrete_path	function	(path, embedding, save)	Animate a point moving along a discrete quadray path.
visualize	animate_simplex	function	(vertices_list, embedding, save)	Animate simplex evolution across iterations.
visualize	plot_ivm_neighbors	function	(embedding, save)	Scatter the 12 IVM neighbor points in 3D.
visualize	plot_partition_tetrahedron	function	(mu, s, a, psi, embedding, save)	Plot the four-fold partition as a labeled tetrahedron in 3D.
visualize	plot_simplex_trace	function	(state, save)	Plot per-iteration diagnostics for Nelder–Mead.