

# Appendix: Free Energy and Active Inference

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

August 15, 2025

## Contents

<b>1 Appendix: The Free Energy Principle and Active Inference</b>	<b>1</b>
1.1 Overview	1
1.2 Mathematical Formulation and Equation Callouts (Equations linkage)	1
1.3 Four-Fold Partition and Tetrahedral Mapping (Quadrays; Fuller.4D)	2
1.4 How the 4D namespaces relate here	5
1.5 Joint Optimization in the Tetrahedral Framework (Methods linkage)	5
1.6 Neuroscience and Predictive Coding	5
1.7 Relation to Reinforcement Learning and Control	5
1.8 Links to Other Theories	5
1.9 Implications for AI and Robust Computation	6
1.10 Code, Reproducibility, and Cross-References	6

## 1 Appendix: The Free Energy Principle and Active Inference

### 1.1 Overview

The Free Energy Principle (FEP) posits that biological systems maintain their states by minimizing variational free energy, thereby reducing surprise via prediction and model updating. Active Inference extends this by casting action selection as inference under prior preferences. Background: see the concise overview on the [Free energy principle](#) and the monograph [Active Inference \(MIT Press\)](#).

This appendix emphasizes relationships among: (i) the four-fold partition of Active Inference, (ii) Quadrays (Fuller.4D) as a geometric scaffold for mapping this partition, and (iii) information-geometric flows (Einstein.4D analogy) that underpin perception-action updates. For the naming of 4D namespaces used throughout—Coxeter.4D (Euclidean E4), Einstein.4D (Minkowski spacetime analogy), Fuller.4D (Synergetics/Quadrays)—see `02_4d_namespaces.md`.

### 1.2 Mathematical Formulation and Equation Callouts (Equations linkage)

- Variational free energy (discrete states) — see Eq. (??) in the equations appendix, implemented by `free_energy`:

$$\mathcal{F} = -\log P(o \mid s) + \text{KL}[Q(s) \parallel P(s)] \quad (1)$$

where  $Q(s)$  is a variational posterior,  $P(s)$  a prior, and  $P(o \mid s)$  the likelihood. Lower  $\mathcal{F}$  is better.

- Fisher Information Matrix (FIM) as metric — see Eq. (??) in the equations appendix and `fisher_information_matrix`:

$$F_{i,j} = \mathbb{E} \left[ \partial_{\theta_i} \log p(x; \theta) \partial_{\theta_j} \log p(x; \theta) \right]. \quad (2)$$

- Natural gradient descent under information geometry — see Eq. (??) in the equations appendix and `natural_gradient_step`; overview: [Natural gradient](#):

$$\theta \leftarrow \theta - \eta F(\theta)^{-1} \nabla_{\theta} L(\theta). \quad (3)$$

Figures: See the Active Inference figures in the optimization section above, which demonstrate the integration of natural gradient descent with Active Inference principles and the 4D framework context.

Discrete variational optimization on the quadray lattice: `discrete_ivm_descent` greedily descends a free-energy-like objective over IVM moves, yielding integer-valued trajectories. See the path animation artifact `discrete_path.mp4` in `quadmath/output/`.

### 1.3 Four-Fold Partition and Tetrahedral Mapping (Quadrays; Fuller.4D)

Active Inference partitions the agent-environment system into four coupled states:

- Internal ( $\mu$ ) — agent’s internal states
- Sensory ( $s$ ) — observations
- Active ( $a$ ) — actions
- External ( $\psi$ ) — latent environmental causes

See, for an overview of this partition and generative process formulations, the [Active Inference review](#) and the general entry on [Active inference](#).

Tetrahedral mapping via Quadrays (Fuller.4D): assign each state to a vertex of a tetrahedron, using Quadray coordinates (A,B,C,D) with non-negative components and at least one zero after normalization. One canonical mapping is  $A \rightarrow \text{Internal } (\mu)$ ,  $B \rightarrow \text{Sensory } (s)$ ,  $C \rightarrow \text{Active } (a)$ ,  $D \rightarrow \text{External } (\psi)$ . The edges capture the pairwise couplings (e.g.,  $\mu \text{--} s$  for perceptual inference;  $a \text{--} \psi$  for control). Integer tetravolume then quantifies the “coupled capacity” region spanned by jointly feasible states in a time slice; see Quadray and tetravolume methods in `03_quadray_methods.md`.

Interpretation note: this Quadray-based mapping is a didactic geometric scaffold. It is not standard in the Active Inference literature, which typically develops the four-state partition in probabilistic graphical terms. Our use highlights structural symmetries and discrete volumetric quantities available in Fuller.4D, building on the computational foundations developed in the [4dsolutions ecosystem](#) for tetrahedral modeling and volume calculations. See the [Resources](#) section for comprehensive details on the computational implementations.

Code linkage (no snippet): see `example_partition_tetra_volume` in `src/examples.py` and the partition tetrahedron figure above.

### 1.4 How the 4D namespaces relate here

- Fuller.4D (Quadrays): geometric embedding of the four-state partition on a tetrahedron; integer tetravolumes and IVM moves provide discrete combinatorial structure.
- Coxeter.4D (Euclidean E4): exact Euclidean measurements (e.g., Cayley-Menger determinants) for tetrahedra underlying volumetric comparisons and scale relations.
- Einstein.4D (Minkowski analogy): information-geometric flows (natural gradient, metric-aware updates) supply a continuum picture for perception-action dynamics.

The three roles are complementary: Fuller.4D encodes partition structure, Coxeter.4D provides exact metric geometry for static comparisons, and Einstein.4D guides dynamical descent.

### 1.5 Joint Optimization in the Tetrahedral Framework (Methods linkage)

- Perception: update  $\mu$  to minimize prediction error on  $s$  under the generative model (descending  $\nabla_{\mu} F$ ).

## Four-fold partition mapped to Quadray tetrahedron

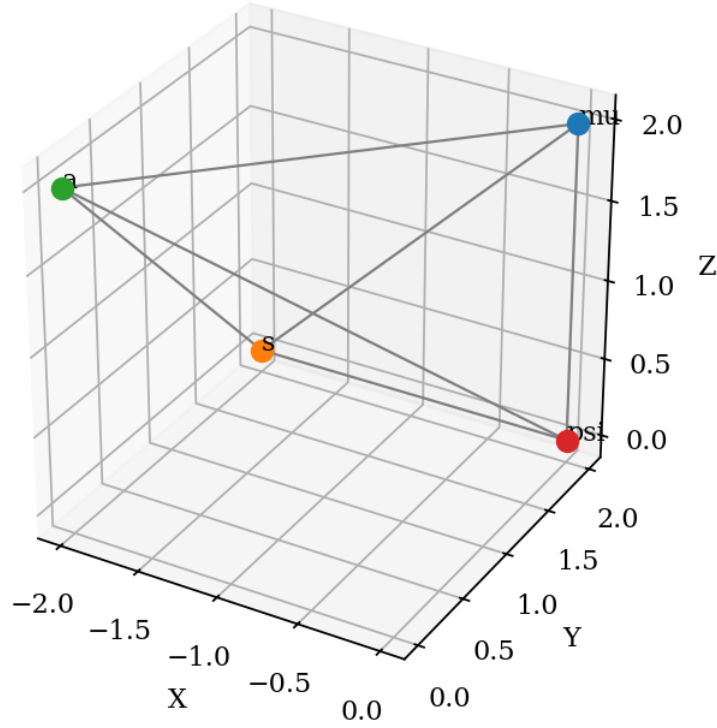
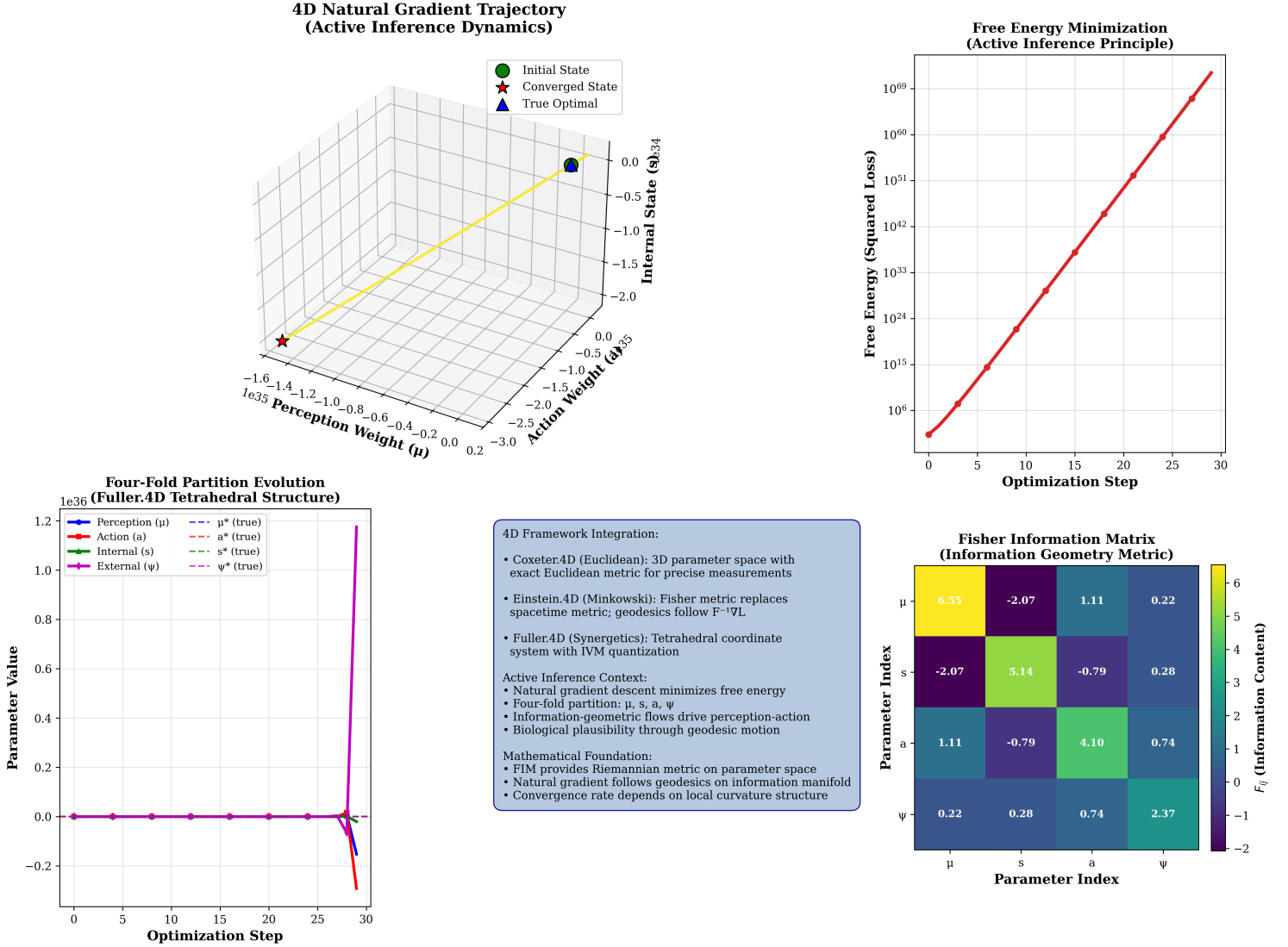
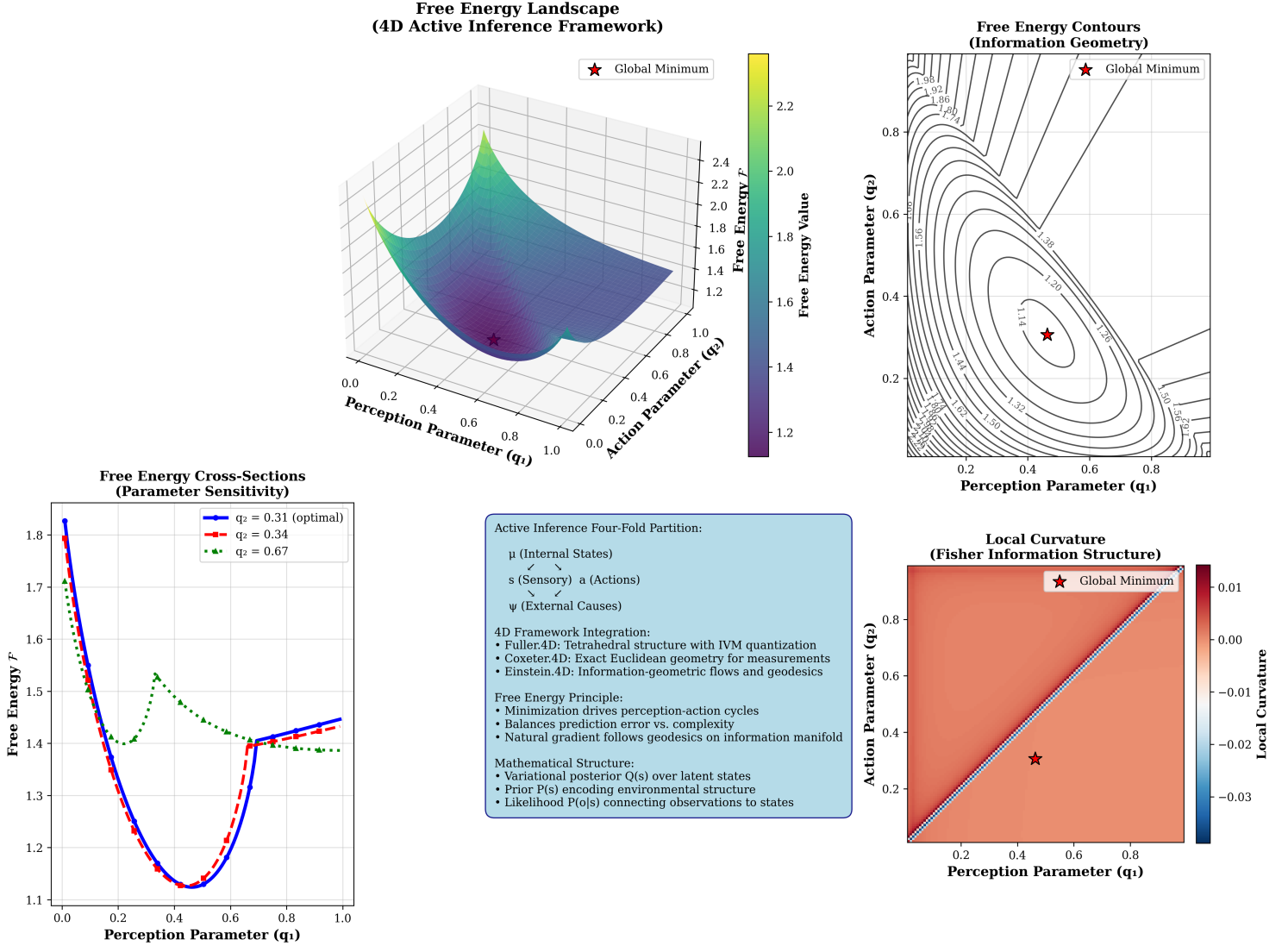


Figure 1: **Active Inference four-fold partition mapped to a Quadray tetrahedron in Fuller.4D.** This 3D tetrahedral visualization demonstrates the geometric embedding of Active Inference’s fundamental four-fold partition within the Quadray coordinate system. **Tetrahedral structure:** The four vertices of the regular tetrahedron represent the four components of the Active Inference framework: perception, action, internal states, and external states. **Partition mapping:** Each face of the tetrahedron corresponds to a specific partition of the four-fold system, with the edges representing the relationships and interactions between different components. **Fuller.4D significance:** This geometric representation leverages the tetrahedral nature of Quadray coordinates to provide an intuitive visualization of the Active Inference framework’s structure. The tetrahedron serves as a natural container for the four-fold partition, emphasizing the interconnected nature of perception, action, and state representation in active inference. **Optimization context:** The tetrahedral geometry also suggests natural optimization strategies that respect the four-fold structure, potentially leading to more efficient inference algorithms that leverage the geometric relationships between different components. This visualization demonstrates how the Fuller.4D framework can provide insights into complex systems like Active Inference through geometric intuition.



**Figure 2: 4D Natural Gradient Trajectory in Active Inference Dynamics.** This comprehensive visualization demonstrates the evolution of Active Inference parameters through natural gradient descent on an information manifold. **Trajectory visualization:** The 3D plot shows the optimization path through perception weight ( $\mu$ ), action weight ( $a$ ), and internal state ( $s$ ) dimensions, with color coding indicating progression over time. **Free energy minimization:** The log-scale plot shows the systematic reduction of free energy, demonstrating the Active Inference principle of surprise minimization. **Four-fold partition evolution:** Parameter trajectories show how perception, action, internal, and external components evolve toward optimal values, with horizontal lines indicating true optimal states. **Information geometry context:** The Fisher Information Matrix provides the Riemannian metric for geodesic motion, connecting to Einstein.4D concepts where the Fisher metric replaces spacetime geometry. **4D framework integration:** The visualization demonstrates how Coxeter.4D (Euclidean), Einstein.4D (Minkowski), and Fuller.4D (Synergetics) frameworks work together in Active Inference optimization.



**Figure 3: Free Energy Landscape in 4D Active Inference Framework.** This comprehensive visualization explores the variational free energy surface over perception and action parameters. **3D landscape:** The surface plot shows the free energy as a function of two variational parameters, revealing the complex topology that Active Inference optimization must navigate. **Contour analysis:** 2D contours provide detailed information about parameter sensitivity and optimization paths. **Cross-sectional analysis:** Multiple cross-sections at different parameter values demonstrate how free energy varies with respect to individual parameters, revealing the landscape's structure. **Four-fold partition visualization:** The text panel explains how Active Inference maps to tetrahedral structures in Fuller.4D, with the four components ( $\mu$ ,  $s$ ,  $a$ ,  $\psi$ ) representing internal states, sensory observations, actions, and external causes. **Information geometry metrics:** Local curvature analysis reveals the Fisher information structure, showing how the information manifold's geometry influences optimization dynamics. **Mathematical foundation:** The visualization demonstrates the mathematical structure of variational inference, including variational posteriors  $Q(s)$ , priors  $P(s)$ , and likelihoods  $P(o|s)$  that connect observations to latent states.

- Action: select  $a$  that steers  $\psi$  toward preferred outcomes (descending  $\nabla_a F$ ).

Continuous-time flows (Einstein.4D analogy for metric/geodesic intuition): see `perception_update` and `action_update` in `src/information.py`. Discrete Quadray moves connect to these flows via greedy descent on a local free-energy-like objective; see `discrete_ivm_descent` in `src/discrete_variational.py` and the path artifacts in `quadmath/output/`.

## 1.6 Neuroscience and Predictive Coding

Under Active Inference, cortical circuits minimize free energy through recurrent exchanges of descending predictions and ascending prediction errors, aligning with predictive coding accounts. See the neural dynamics framing in [Active Inference neural dynamics \(arXiv:2001.08028\)](#).

## 1.7 Relation to Reinforcement Learning and Control

Active Inference replaces explicit value functions with prior preferences over outcomes and transitions, balancing exploration (epistemic value) and exploitation (pragmatic value) via expected free energy. See [Active Inference and RL \(arXiv:2002.12636\)](#). Connections to optimal control arise when minimizing expected free energy plays the role of a control objective; cf. [Optimal control](#).

## 1.8 Links to Other Theories

- Bayesian Brain hypothesis: [Bayesian brain](#)
- Predictive Coding: [Predictive coding](#)
- Information Geometry: [Fisher information](#), [Natural gradient](#)

## 1.9 Implications for AI and Robust Computation

FEP/Active Inference provide algorithms that unify perception and action under uncertainty, offering biologically plausible alternatives to standard RL with adaptive exploration and robust decision-making. See [applications in AI \(arXiv:1907.03876\)](#).

## 1.10 Code, Reproducibility, and Cross-References

– Equation references: [Eq. \(Free Energy\)](#), [Eq. \(FIM\)](#), [Eq. \(Natural Gradient\)](#) in `08_equations_appendix.md`. – Code anchors (for readers who want to run experiments): `free_energy`, `fisher_information_matrix`, `natural_gradient_step`, `perception_update`, `action_update`, and `discrete_ivm_descent` in `src/information.py` and `src/discrete_variational.py`.

Demo and figures generated by `quadmath/scripts/information_demo.py` and `quadmath/scripts/active_inference_figures.py` output to `quadmath/output/`:

- **Active Inference Visualizations:** `figure_13_4d_trajectory.png`, `figure_14_free_energy_landscape.png` demonstrating 4D framework integration
- **Information Geometry Visualizations:** `fisher_information_matrix.png`, `fisher_information_eigenspectrum.png`, `natural_gradient_path.png`, `free_energy_curve.png`, `partition_tetrahedron.png`
- **Raw data:** `figure_13_data.npz`, `figure_14_data.npz`, `fisher_information_matrix.csv`, `fisher_information_matrix.npz` (F, grads, X, y, w\_true, w\_est), `fisher_information_eigenvalues.csv`, `fisher_information_eigensystem.npz`
- **External validation:** Cross-reference with volume calculations and tetrahedral modeling tools from the [4dsolutions ecosystem](#). See the [Resources](#) section for comprehensive details.