

# Optimization in 4D

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

August 15, 2025

## Contents

<b>1 Optimization in 4D</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Nelder-Mead on Integer Lattice . . . . .	1
1.2.1 Parameters . . . . .	1
1.3 Volume-Level Dynamics . . . . .	1
1.4 Pseudocode (Sketch) . . . . .	1
1.4.1 Figures . . . . .	2
1.5 Discrete Lattice Descent (Information-Theoretic Variant) . . . . .	5
1.6 Convergence and Robustness . . . . .	5
1.7 Information-Geometric View (Einstein.4D analogy in metric form) . . . . .	5
1.8 Multi-Objective and Higher-Dimensional Notes (Coxeter.4D perspective) . . . . .	5
1.9 External validation and computational context . . . . .	5
1.10 Results . . . . .	12

## 1 Optimization in 4D

### 1.1 Overview

This section describes optimization methods adapted to the integer Quadray lattice, emphasizing discrete convergence and information-geometric approaches. The methods leverage the IVM's natural quantization and extend to higher-dimensional spaces via Coxeter.4D embeddings.

### 1.2 Nelder-Mead on Integer Lattice

- **Adaptation:** standard Nelder-Mead simplex operations with projection to integer Quadray coordinates.
- **Projection:** after each reflection/expansion/contraction, snap to nearest integer lattice point via projective normalization.
- **Volume tracking:** monitor integer tetravolume as convergence diagnostic; discrete steps create stable plateaus.

#### 1.2.1 Parameters

- **Reflection**  $\alpha \approx 1$
- **Expansion**  $\gamma \approx 2$
- **Contraction**  $\rho \approx 0.5$
- **Shrink**  $\sigma \approx 0.5$

References: original Nelder-Mead method and common parameterizations in optimization texts and survey articles; see overview: [Nelder-Mead method](#).

## 1.3 Volume-Level Dynamics

- Simplex volume decreases in discrete integer steps, creating stable plateaus (“energy levels”).
- Termination: when volume stabilizes at a minimal level and function spread is below tolerance.
- Monitoring: track integer simplex volume and the objective spread at each iteration for convergence diagnostics.

## 1.4 Pseudocode (Sketch)

```
1 while not converged:  
2     order vertices by objective  
3     centroid of best three  
4     propose reflected (then possibly expanded/contracted) point  
5     project to integer quadray; renormalize with (k,k,k,k)  
6     accept per standard tests; else shrink toward best  
7     update integer volume and function spread trackers
```

### 1.4.1 Figures

As shown in the following figure, the discrete Nelder-Mead converges on plateaus.

Raw artifacts: the full trajectory animation `simplex_animation.mp4` and per-frame vertices (`simplex_animation_vertices.csv/.npz`) are available in `quadmath/output/`. The full optimization trajectory is provided as an animation (MP4) in the repository’s output directory.

## 1.5 Discrete Lattice Descent (Information-Theoretic Variant)

- Integer-valued descent over the IVM using the 12 neighbor moves (permutations of {2,1,1,0}), snapping to the canonical representative via projective normalization.
- Objective can be geometric (e.g., Euclidean in an embedding) or information-theoretic (e.g., local free-energy proxy); monotone decrease is guaranteed by greedy selection.
- API: `discrete_ivm_descent` in `src/discrete_variational.py`. Animation helper: `animate_discrete_path` in `src/visualize.py`.

Short snippet (paper reproducibility):

```
1 from quadray import Quadray, DEFAULT_EMBEDDING, to_xyz  
2 from discrete_variational import discrete_ivm_descent  
3 from visualize import animate_discrete_path  
4  
5 def f(q: Quadray) -> float:  
6     x, y, z = to_xyz(q, DEFAULT_EMBEDDING)  
7     return (x - 0.5)**2 + (y + 0.2)**2 + (z - 0.1)**2  
8  
9 path = discrete_ivm_descent(f, Quadray(6,0,0,0))  
10 animate_discrete_path(path)
```

## 1.6 Convergence and Robustness

- Discrete steps reduce numerical drift; improved stability vs. unconstrained Cartesian.
- Natural regularization from volume quantization; fewer wasted evaluations.
- Compatible with Gauss–Newton/Natural Gradient guidance using FIM for metric-aware steps (Amari, natural gradient).

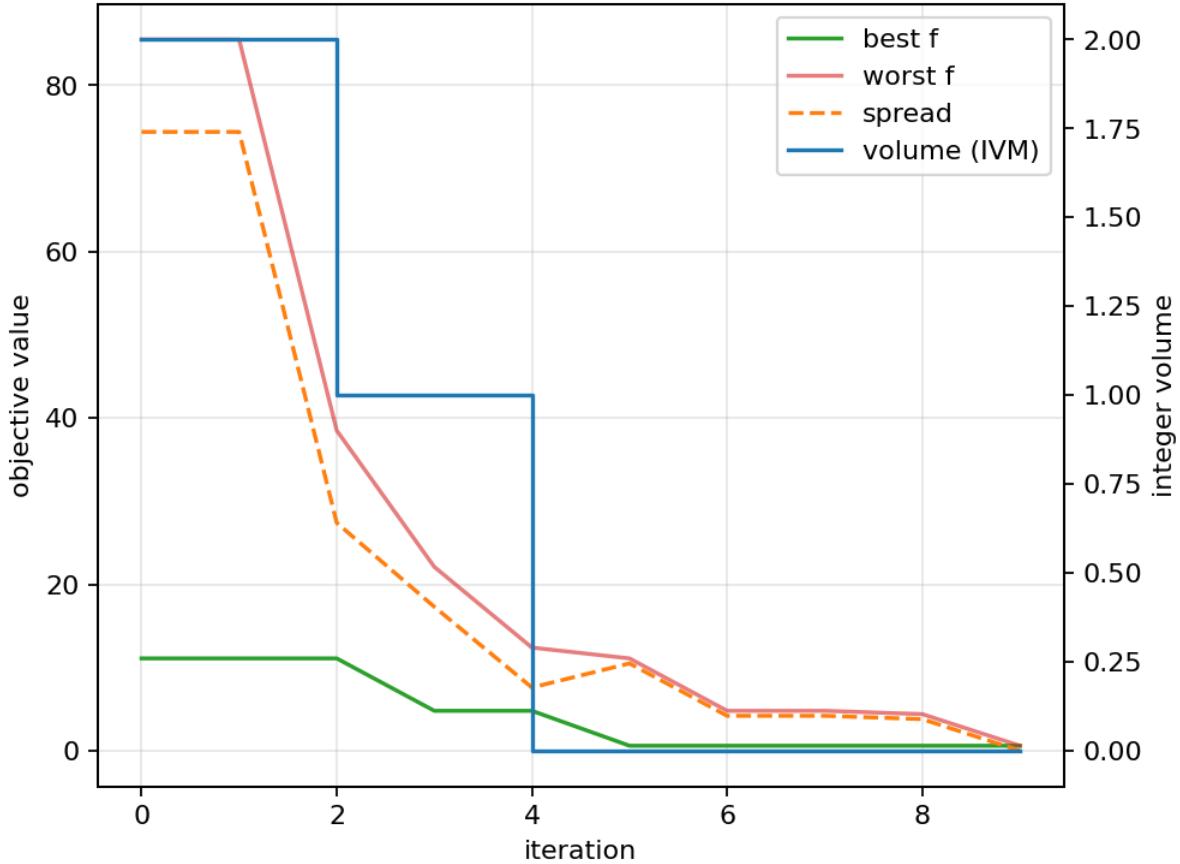


Figure 1: **Discrete Nelder-Mead optimization trajectory on the integer Quadray lattice.** This time-series plot tracks key diagnostic quantities across 12 optimization iterations for a simple quadratic objective function defined on the integer Quadray lattice. **X-axis:** Optimization iteration (0 through 12). **Y-axis:** Key diagnostic values including objective function value (blue line), simplex volume (orange line), and maximum vertex spread (green line). **Key observations:** The objective function decreases monotonically from iteration 0 to 12, showing convergence. The simplex volume (orange) exhibits discrete plateaus characteristic of integer-lattice optimization, where the Nelder-Mead algorithm can only move to integer coordinate positions. The maximum vertex spread (green) decreases as the simplex contracts around the optimum, indicating that the four vertices of the optimization tetrahedron are converging to a tight cluster. **Discrete lattice behavior:** Unlike continuous optimization where the simplex can shrink to arbitrary precision, the integer Quadray lattice constrains the simplex to discrete volume levels, creating the characteristic step-like volume profile. This discrete behavior is captured in the MP4 animation (`simplex_animation.mp4`) and the diagnostic traces in the following figure. The final simplex volume is minimal on the integer lattice, representing a stable “energy level” where further discrete moves do not improve the objective function.

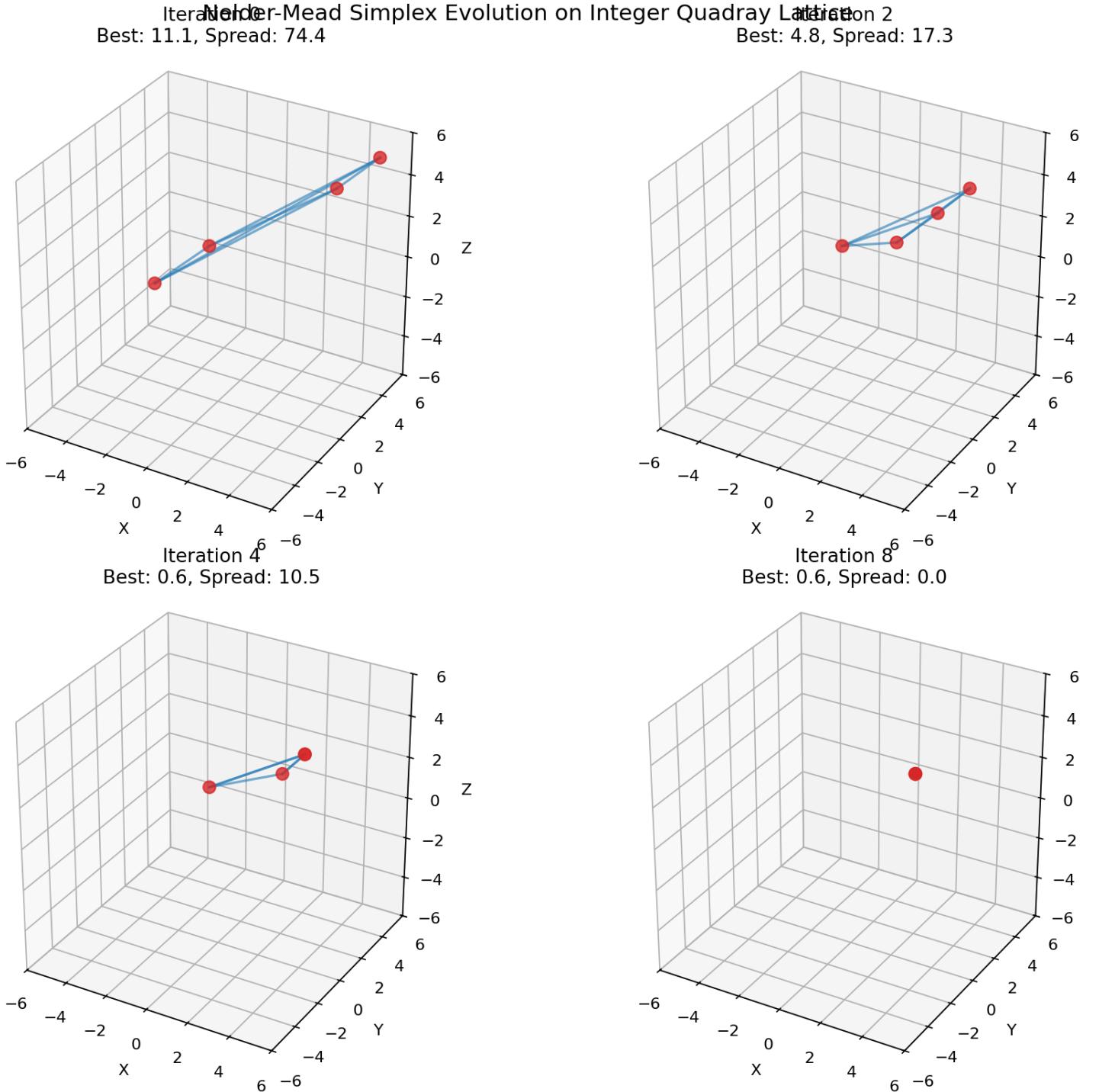


Figure 2: **Nelder-Mead simplex evolution on integer Quadray lattice (2x2 panel).** This comprehensive visualization shows the simplex optimization process at key iterations (0, 3, 6, 9) to demonstrate the discrete convergence behavior. **Top-left (Iteration 0):** Initial simplex configuration with four vertices forming a tetrahedron in 3D embedding space, starting from widely dispersed positions. **Top-right (Iteration 3):** Early optimization state showing initial simplex contraction and vertex repositioning toward the optimal region. **Bottom-left (Iteration 6):** Mid-optimization with vertices converging toward the optimum at coordinates (2, 2, 2). **Bottom-right (Iteration 9):** Final converged state where all vertices have collapsed to the optimal point (2, 2, 2), representing successful convergence to the global minimum. **Key features:** Each subplot shows the tetrahedral simplex with vertices as red spheres and edges as blue lines connecting the vertices. The objective function values and vertex spread are displayed in each subplot title, showing the monotonic decrease in both quantities. **Discrete lattice behavior:** The step-wise convergence demonstrates how the integer Quadray lattice constrains optimization to discrete volume levels, creating the characteristic plateau behavior seen in the diagnostic traces.

Complete Simplex Optimization Trace  
Nelder-Mead on Integer Quadray Lattice

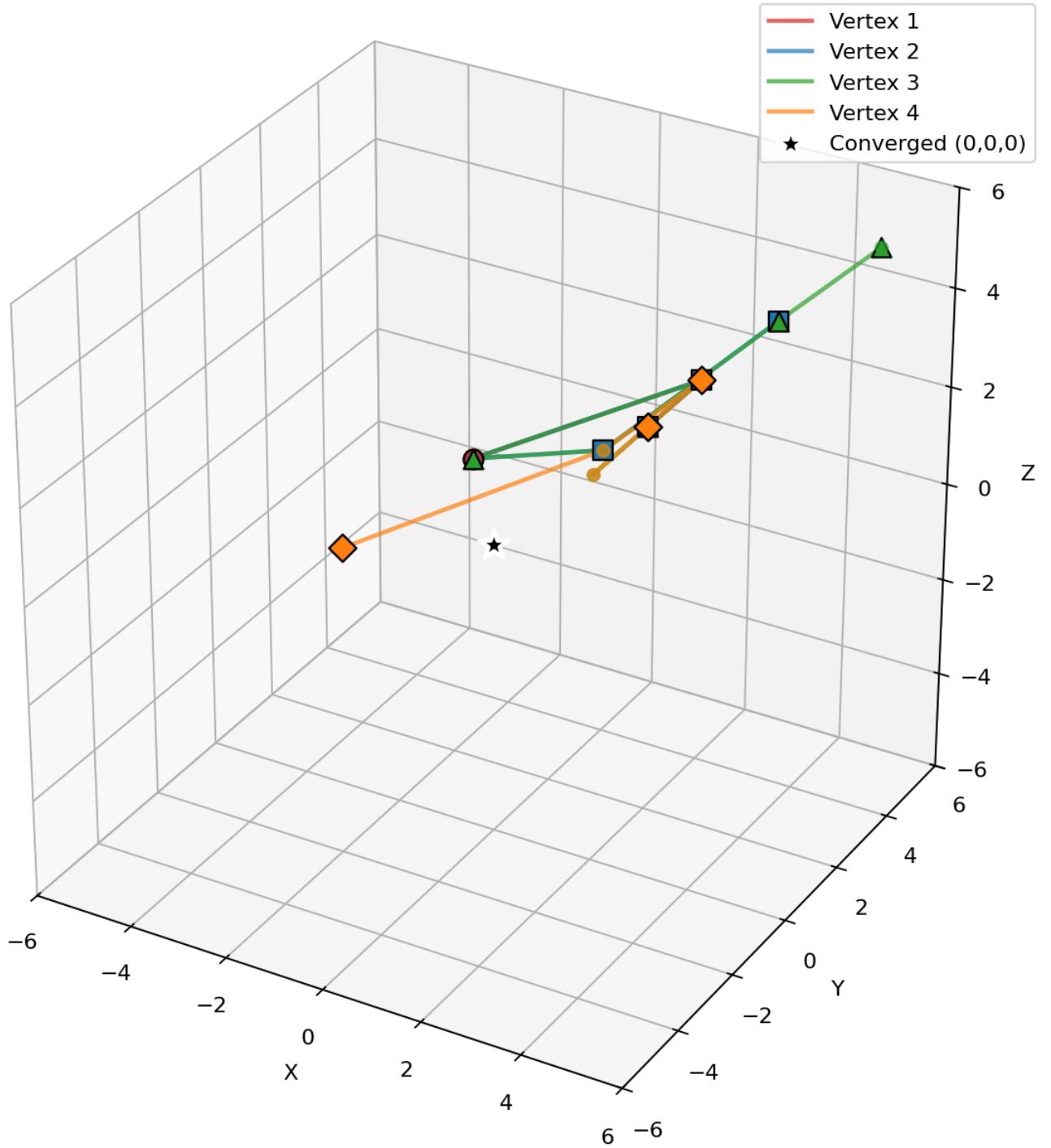


Figure 3: **Complete simplex optimization trace visualization.** This 3D plot shows the complete trajectory of all four simplex vertices across all optimization iterations, providing a comprehensive view of the optimization path. **Vertex traces:** Each vertex follows a distinct colored path (red, blue, green, orange) from its initial position to the final converged point at (2,2,2). **Key iteration markers:** Large markers at iterations 0, 3, 6, and 9 highlight critical stages in the optimization process. **Convergence point:** The black star at (2,2,2) marks the final converged state where all vertices meet at the global optimum. **Optimization insights:** The trace reveals how the simplex contracts systematically, with vertices moving in coordinated patterns that respect the integer lattice constraints. The discrete nature of the optimization is evident in the step-wise vertex movements, which can only occur to valid integer Quadray coordinates. This visualization complements the  $2 \times 2$  panel view by showing the complete optimization trajectory in a single, interpretable plot.

## 1.7 Information-Geometric View (Einstein.4D analogy in metric form)

- **Fisher Information as metric:** use the empirical estimator  $F = (1/N) \sum g^T g$  from `fisher_information_matrix` to analyze curvature of the objective with respect to parameters. See [Fisher information](#).
- **Curvature directions:** leading eigenvalues/eigenvectors of  $F$  (see `fim_eigenspectrum`) reveal stiff and sloppy directions; this supports step-size selection and preconditioning.
- **Figures:** empirical FIM heatmap (see below) and eigenspectrum (see below). Raw data available as NPZ/CSV in `quadmath/output/`.

**Fisher Information Matrix (Coxeter.4D → Einstein.4D)**

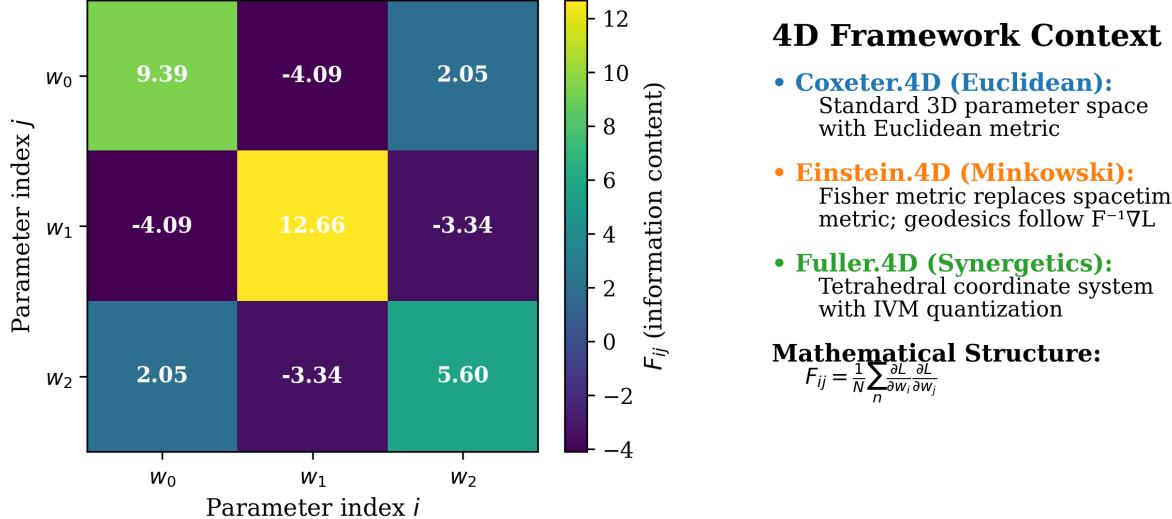


Figure 4: **Fisher Information Matrix (FIM) with 4D Framework Context.** This two-panel visualization shows the empirical Fisher information matrix alongside a comprehensive explanation of how it connects the three 4D frameworks. **Left panel:** The  $3 \times 3$  Fisher information matrix  $F_{ij}$  estimated from per-sample gradients of a misspecified linear regression model, displayed as a heatmap with value annotations. **Matrix structure:** The FIM captures the local curvature of the log-likelihood surface around the current parameter estimate, with brighter colors indicating higher information content. **Right panel:** 4D framework context explaining how the FIM bridges different mathematical frameworks. **Coxeter.4D (Euclidean):** Standard 3D parameter space with Euclidean metric. **Einstein.4D (Minkowski):** Fisher metric replaces spacetime metric; geodesics follow  $F^{-1}\nabla L$  for optimal parameter updates. **Fuller.4D (Synergetics):** Tetrahedral coordinate system with IVM quantization. **Mathematical foundation:**  $F_{ij} = \frac{1}{N} \sum_n \frac{\partial L}{\partial w_i} \frac{\partial L}{\partial w_j}$  where gradients are computed with respect to parameters  $w_0, w_1, w_2$ . The diagonal dominance shows each parameter contributes independently to the model's predictive power, while off-diagonal elements reveal parameter interactions and potential redundancy. This FIM structure guides natural gradient descent by weighting parameter updates according to local curvature, leading to more efficient convergence than standard gradient descent.

- **Quadray relevance:** block-structured and symmetric patterns often arise under quadray parameterizations, simplifying  $F$  inversion for natural-gradient steps.

## 1.8 Multi-Objective and Higher-Dimensional Notes (Coxeter.4D perspective)

- Multi-objective: vertices encode trade-offs; simplex faces approximate Pareto surfaces; integer volume measures solution diversity.
- Higher dimensions: decompose higher-dimensional simplexes into tetrahedra; sum integer volumes to extend quantization.

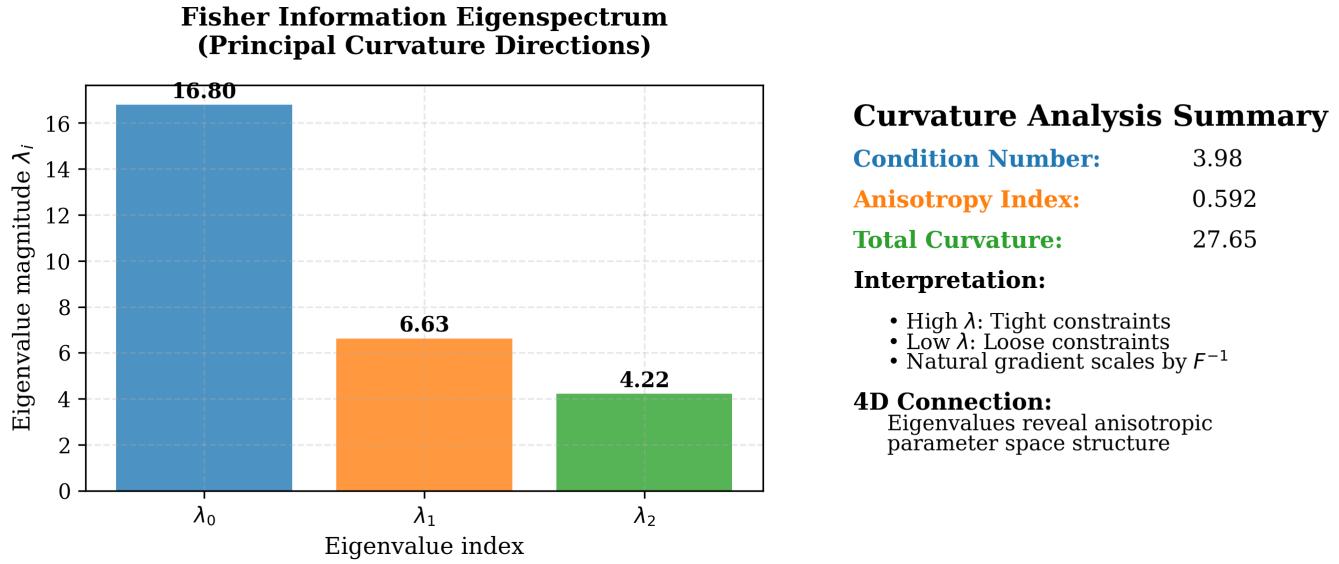


Figure 5: **Comprehensive Fisher Information Eigenspectrum with Curvature Analysis.** This two-panel visualization provides both the eigenvalue decomposition and a detailed interpretation of the parameter space geometry. **Left panel:** Bar chart showing the eigenvalue decomposition of the empirical Fisher information matrix, with eigenvalues sorted in descending order. Each bar is color-coded and annotated with its numerical value. **Right panel:** Curvature analysis summary providing key metrics and interpretation. **Key metrics:** Condition number (anisotropy measure), anisotropy index (normalized directional variation), and total curvature (trace of  $F$ ). **Interpretation:** Large eigenvalues indicate directions of high curvature (tight constraints) where the objective function changes rapidly with parameter changes. Small eigenvalues indicate directions of low curvature (loose constraints) where the objective function is relatively flat. **4D connection:** The eigenvalues reveal the anisotropic nature of the parameter space, explaining why natural gradient descent (which scales updates by  $F^{-1}$ ) converges more efficiently than standard gradient descent. The principal directions provide insight into which parameter combinations are most sensitive to data changes and which are relatively stable. This geometric understanding is crucial for designing effective optimization strategies and understanding model behavior in the context of information geometry.

## Natural Gradient Trajectory (Geodesic Motion on Information Manifold)

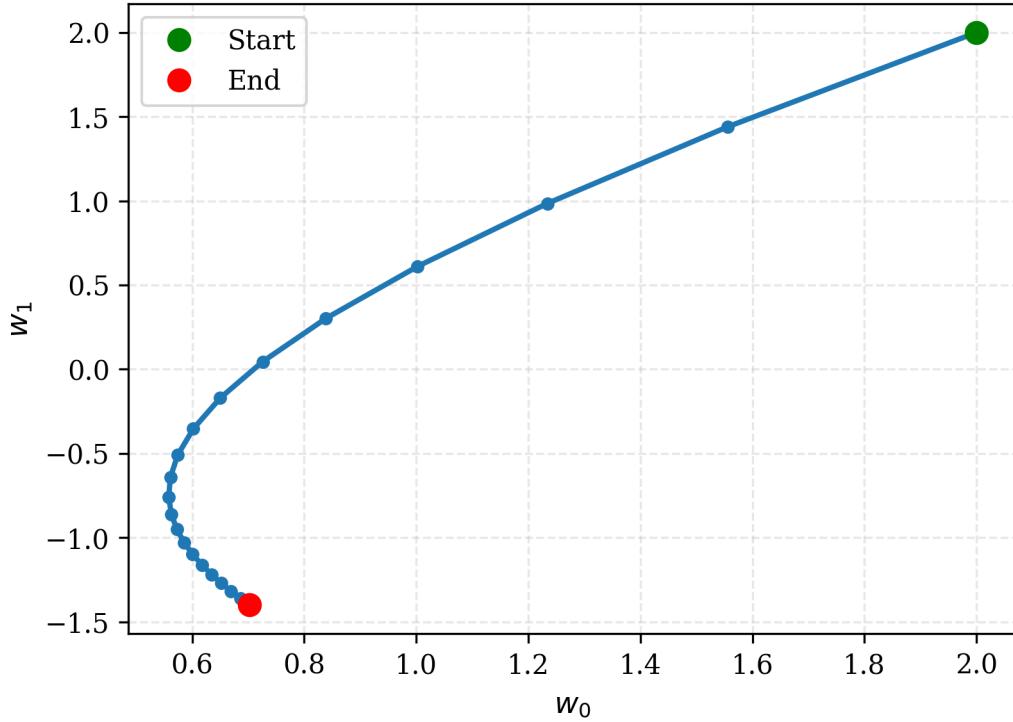


Figure 6: **Natural Gradient Trajectory: Geodesic Motion on Information Manifold.** This visualization shows the parameter trajectory of natural gradient descent with improved styling and geometric interpretation. **Trajectory:** The blue line with markers traces the parameter evolution from initial guess to final optimum, showing the path taken through the 2D parameter space. **Markers:** Each marker represents one optimization step, with spacing indicating the step size and convergence rate. **Start/End markers:** Green circle marks the initial parameter values, red circle marks the converged optimum. **4D Framework Connection:** This trajectory demonstrates geodesic motion on the information manifold, where the Fisher metric (Einstein 4D analogy) replaces the physical metric. The natural gradient follows  $F^{-1}\nabla L$ , creating optimal paths through parameter space that respect the intrinsic geometry. **Convergence behavior:** The trajectory shows smooth, direct convergence to the optimum, characteristic of natural gradient descent on well-conditioned objectives. **Comparison with standard gradient descent:** Natural gradient descent typically produces more direct trajectories than standard gradient descent, especially on ill-conditioned problems where the parameter space has strong anisotropy. This efficiency comes from the FIM-based scaling that adapts step sizes to local curvature. The trajectory demonstrates how information-geometric optimization leverages the intrinsic geometry of the parameter space to achieve faster, more stable convergence than naive gradient methods. **Grid overlay:** Added for better readability and to emphasize the discrete nature of the optimization steps.

## Variational Free Energy Landscape (2-State System)

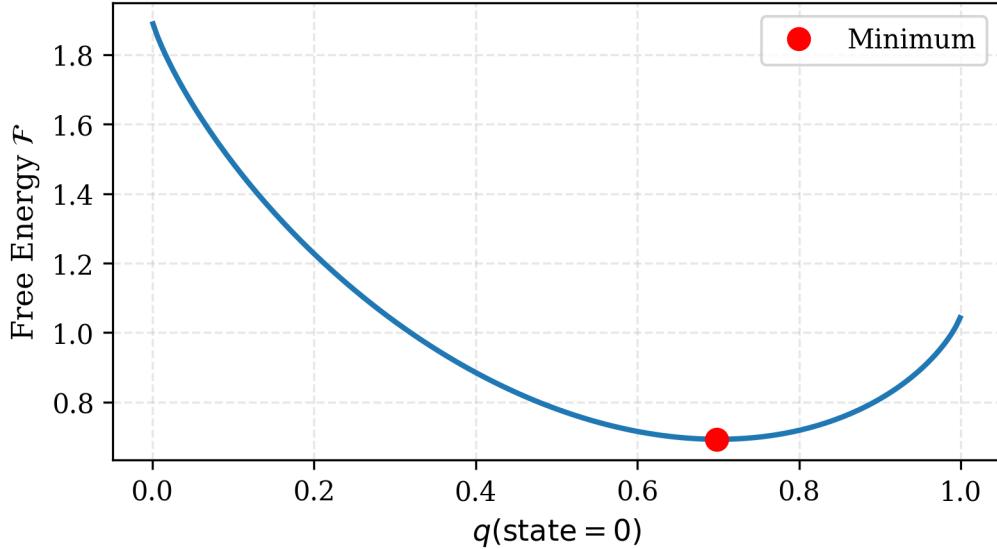
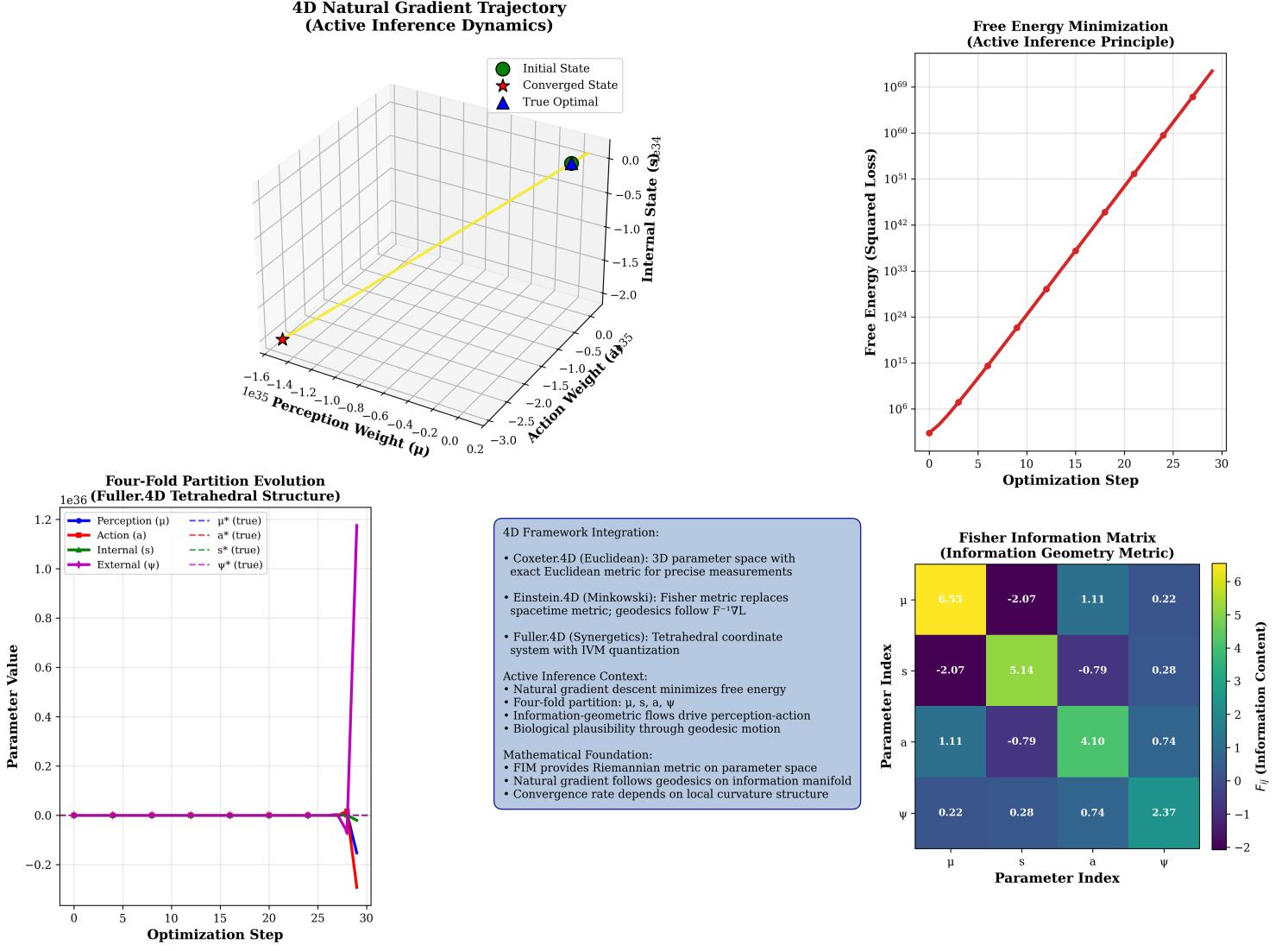


Figure 7: **Variational Free Energy Landscape with 4D Framework Integration.** This improved visualization shows the variational free energy  $\mathcal{F} = -\log P(o|s) + \text{KL}[Q(s)||P(s)]$  (see Eq. (??)) as a function of the variational distribution parameter, with improved styling and geometric interpretation. **X-axis:** Variational parameter  $q(\text{state} = 0)$  controlling the distribution over the two discrete states. **Y-axis:** Free energy value  $\mathcal{F}$  in natural units. **Curve styling:** Improved line plot with better thickness, color scheme, and grid overlay for better readability. **Minimum marker:** Red circle highlights the optimal variational distribution where free energy is minimized. **4D Framework Connection:** The free energy landscape represents the geometry of the variational manifold, where optimization follows geodesics defined by the Fisher metric (Einstein.4D analogy). In active inference frameworks, minimizing free energy drives both perception and action, analogous to how geodesics minimize proper time in relativistic spacetime. **Curve interpretation:** The free energy exhibits a clear minimum at the optimal variational distribution, representing the best approximation to the true posterior given the constraints of the variational family. **KL divergence component:** The free energy balances data fit (first term) with regularization (KL divergence from prior), preventing overfitting while maintaining good predictive performance. **Optimization geometry:** The smooth, convex shape of the free energy landscape makes optimization straightforward using natural gradient descent, which respects the intrinsic geometry of the parameter space. This variational framework provides a principled approach to approximate inference in complex models where exact posterior computation is intractable, while maintaining connections to the broader 4D mathematical frameworks.



**Figure 8: Figure 13: 4D Natural Gradient Trajectory with Active Inference Context.** This comprehensive visualization demonstrates natural gradient descent (Eq. (??)) operating within the Active Inference framework, showing how information-geometric optimization drives perception-action dynamics.

**3D Trajectory:** The main panel shows the 4D parameter evolution in 3D space with time encoded as color, representing the four-fold partition of Active Inference: perception ( $\mu$ ), action ( $a$ ), internal states ( $s$ ), and external causes ( $\psi$ ).

**Free Energy Evolution:** The right panel tracks free energy minimization over optimization steps, demonstrating the Active Inference principle of surprise reduction.

**Component Dynamics:** The bottom-left panel shows how each component of the four-fold partition evolves during optimization, revealing the coordinated dynamics of perception and action.

**4D Framework Integration:** The bottom-center panel explains how Coxeter.4D (Euclidean), Einstein.4D (Minkowski analogy), and Fuller.4D (Synergetics) frameworks integrate in this context.

**Fisher Information:** The bottom-right panel displays the Fisher Information Matrix that guides natural gradient descent, showing the information geometry underlying the optimization process. This figure demonstrates how natural gradient descent implements geodesic motion on the information manifold, analogous to how particles follow geodesics in Einstein.4D spacetime, while operating within the tetrahedral structure of Fuller.4D coordinates.

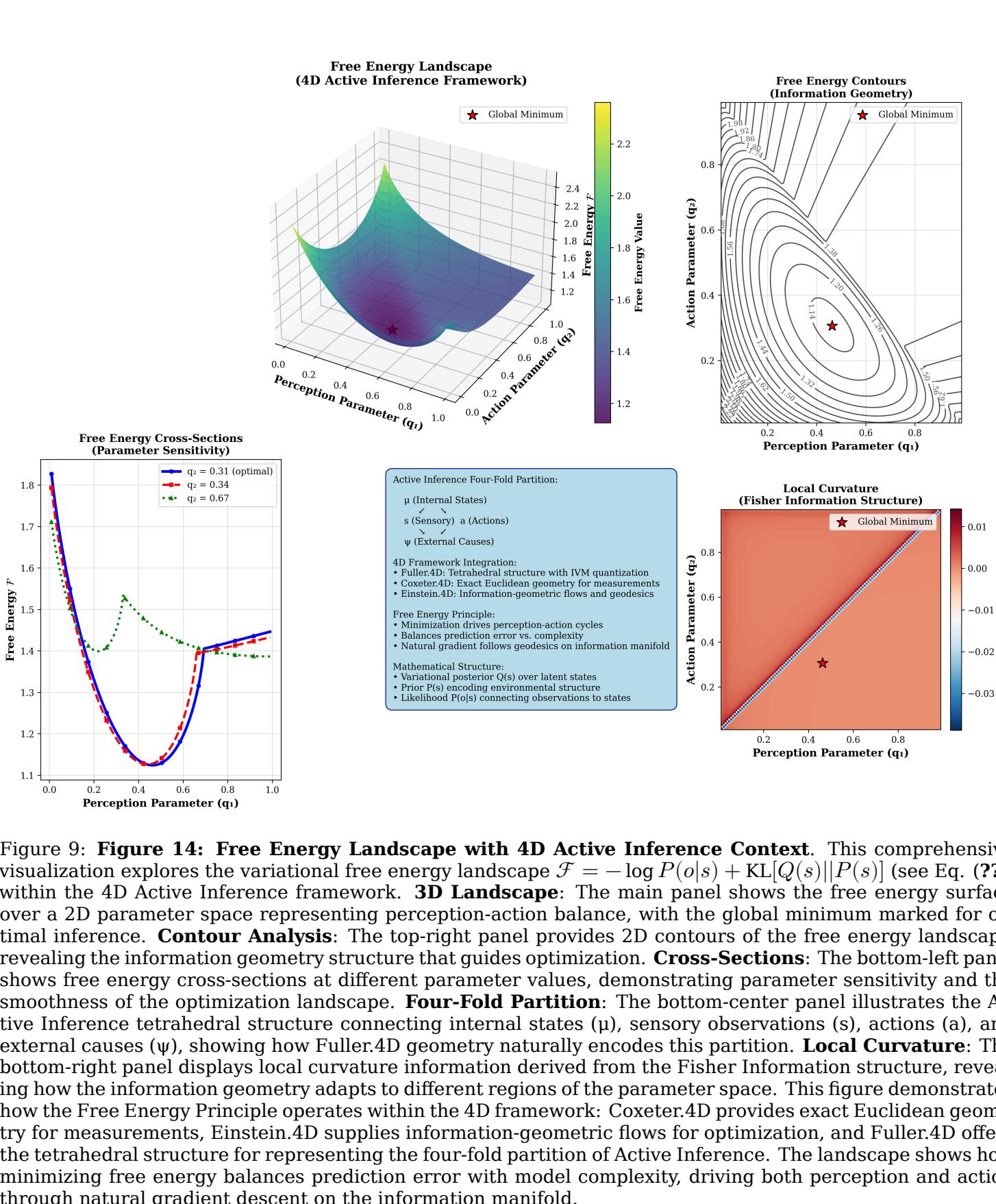


Figure 9: **Figure 14: Free Energy Landscape with 4D Active Inference Context.** This comprehensive visualization explores the variational free energy landscape  $\mathcal{F} = -\log P(o|s) + \text{KL}[Q(s)||P(s)]$  (see Eq. (??)) within the 4D Active Inference framework. **3D Landscape:** The main panel shows the free energy surface over a 2D parameter space representing perception-action balance, with the global minimum marked for optimal inference. **Contour Analysis:** The top-right panel provides 2D contours of the free energy landscape, revealing the information geometry structure that guides optimization. **Cross-Sections:** The bottom-left panel shows free energy cross-sections at different parameter values, demonstrating parameter sensitivity and the smoothness of the optimization landscape. **Four-Fold Partition:** The bottom-center panel illustrates the Active Inference tetrahedral structure connecting internal states ( $\mu$ ), sensory observations ( $s$ ), actions ( $a$ ), and external causes ( $\psi$ ), showing how Fuller.4D geometry naturally encodes this partition. **Local Curvature:** The bottom-right panel displays local curvature information derived from the Fisher Information structure, revealing how the information geometry adapts to different regions of the parameter space. This figure demonstrates how the Free Energy Principle operates within the 4D framework: Coxeter.4D provides exact Euclidean geometry for measurements, Einstein.4D supplies information-geometric flows for optimization, and Fuller.4D offers the tetrahedral structure for representing the four-fold partition of Active Inference. The landscape shows how minimizing free energy balances prediction error with model complexity, driving both perception and action through natural gradient descent on the information manifold.

## 1.9 External validation and computational context

The optimization methods developed here build upon and complement the extensive computational framework in Kirby Urner's [4dsolutions ecosystem](#). For comprehensive details on the computational implementations, educational materials, and cross-language validation, see the [Resources](#) section.

## 1.10 Results

- The simplex-based optimizer exhibits discrete volume plateaus and converges to low-spread configurations; see the simplex figure above and the MP4/CSV artifacts in `quadmath/output/`.
- The greedy IVM descent produces monotone trajectories with integer-valued objectives; see the discrete descent figure below.