

Optimization in 4D

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

August 15, 2025

Contents

1 Optimization in 4D	1
1.1 Overview	1
1.2 Nelder-Mead on Integer Lattice	1
1.2.1 Parameters	1
1.3 Volume-Level Dynamics	2
1.4 Pseudocode (Sketch)	2
1.4.1 Figures	2
1.5 Discrete Lattice Descent (Information-Theoretic Variant)	2
1.6 Convergence and Robustness	2
1.7 Information-Geometric View (Einstein.4D analogy in metric form)	6
1.8 Multi-Objective and Higher-Dimensional Notes (Coxeter.4D perspective)	6
1.9 External validation and computational context	10
1.10 Results	10

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; the previous figure summarizes the scaling behavior used in volume diagnostics.

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).

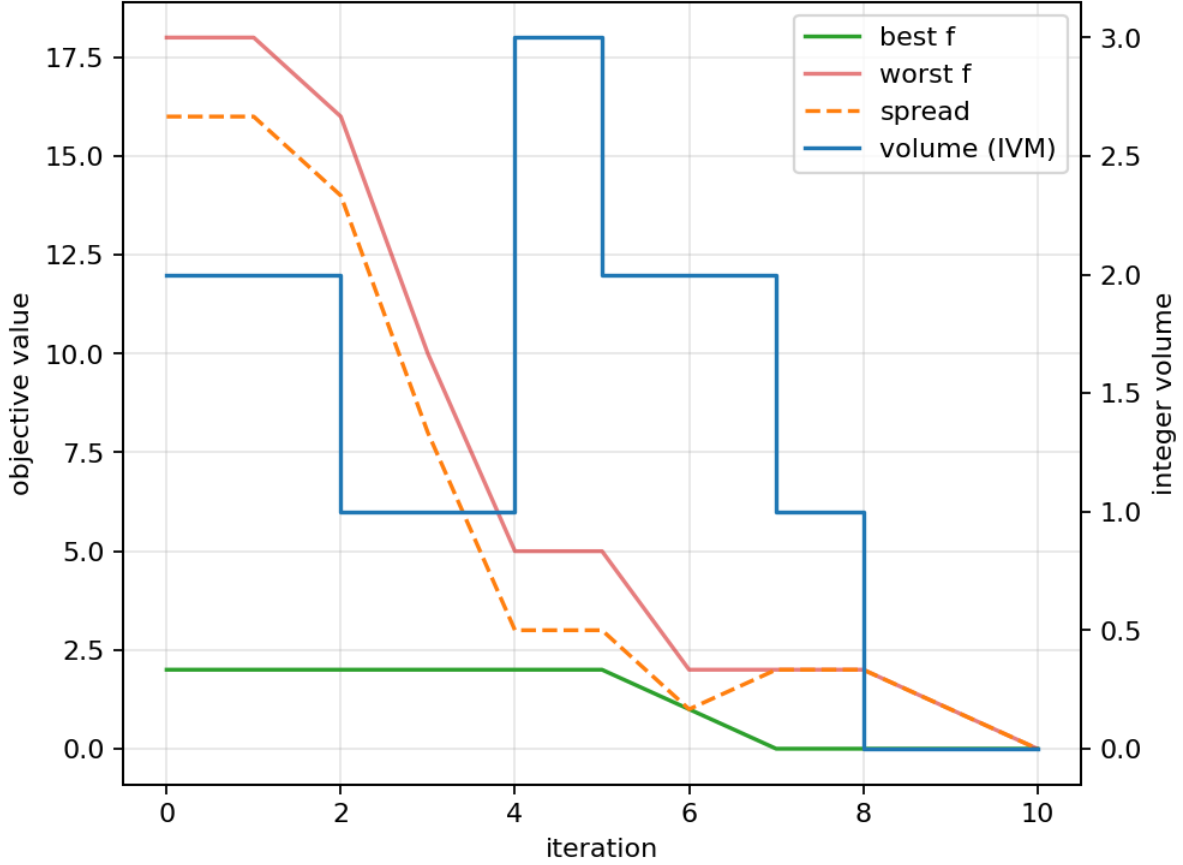


Figure 1: **Discrete Nelder-Mead optimization trajectory on the integer Quadray lattice.** This time-series plot tracks key diagnostic quantities across 12 optimization iterations for a simple quadratic objective function defined on the integer Quadray lattice. **X-axis:** Optimization iteration (0 through 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.

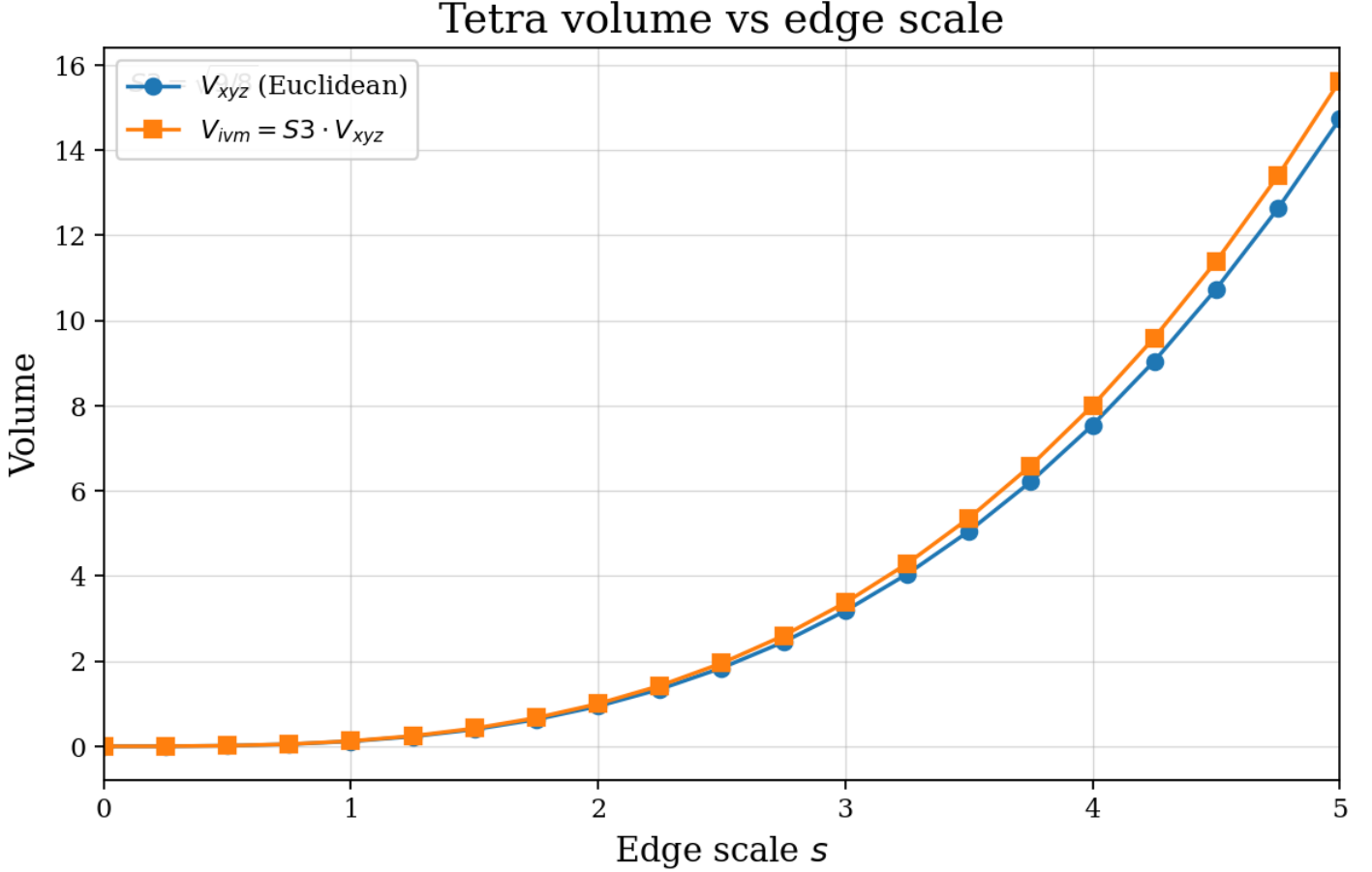


Figure 2: **Tetrahedron volume scaling relationships: Euclidean vs IVM unit conventions.** This plot demonstrates the mathematical relationship between edge length scaling and tetrahedron volume under both Euclidean (XYZ) and IVM (synergetics) unit conventions. **X-axis:** Edge length scaling factor (0.5 to 2.0). **Y-axis:** Tetrahedron volume in respective units. **Blue line (Euclidean):** Volume scales as the cube of edge length, following the standard $V = \frac{\sqrt{2}}{12} \cdot L^3$ relationship for regular tetrahedra. **Orange line (IVM):** Volume scales as the cube of edge length but in IVM tetra-units, following $V_{ivm} = \frac{1}{8} \cdot L^3$ where the regular tetrahedron with unit edge has volume 1/8. **Key insight:** The ratio between these two scaling laws is the synergetics factor $S3 = \sqrt{9/8} \approx 1.06066$, which converts between Euclidean and IVM volume conventions. **Discrete optimization context:** When working on the integer Quadray lattice, this scaling relationship helps diagnose whether volume changes are due to geometric scaling or discrete lattice effects. The plot shows that both conventions preserve the cubic scaling relationship, but with different fundamental units reflecting the different geometric assumptions of Coxeter.4D (Euclidean) versus Fuller.4D (synergetics) frameworks.

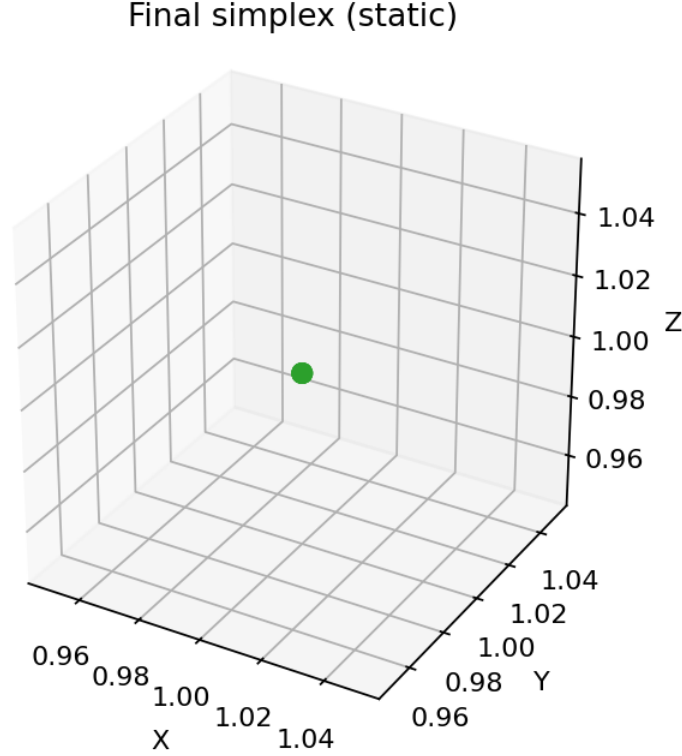


Figure 3: **Final converged simplex configuration in 3D embedding space.** This 3D scatter plot shows the four vertices of the Nelder-Mead simplex after 12 iterations of discrete optimization on the integer Quadray lattice. **Points:** Four colored spheres representing the final simplex vertices, each positioned at integer Quadray coordinates projected to 3D space via the default embedding matrix. **Colors:** Each vertex has a distinct color (blue, orange, green, red) for easy identification. **Optimization context:** These vertices represent the final state of the discrete Nelder-Mead algorithm after converging to a local optimum on the integer lattice. The tight clustering of vertices indicates successful convergence, with the simplex having contracted around the optimal point. **Lattice constraints:** All vertex positions correspond to integer Quadray coordinates, demonstrating the discrete nature of the optimization. The final simplex volume is minimal on the integer lattice, representing a stable configuration where further discrete moves do not improve the objective function. This visualization complements the time-series animation (`simplex_animation.mp4`) and the diagnostic traces in the previous 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.

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

- **Fisher Information as metric:** use the empirical estimator $F = (1/N) \sum g g^{\top}$ 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/`.

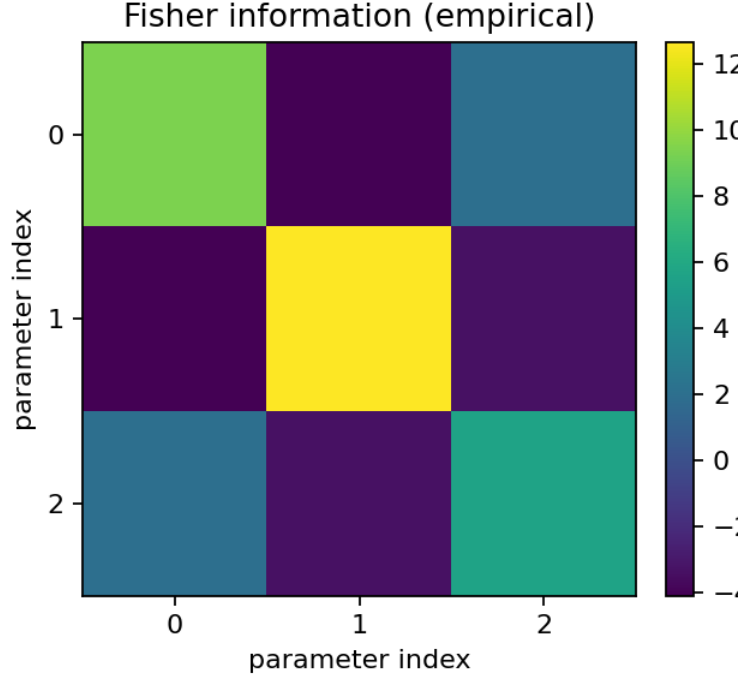


Figure 4: **Empirical Fisher Information Matrix (FIM) for noisy linear regression.** This heatmap visualizes the 3×3 Fisher information matrix F_{ij} estimated from per-sample gradients of a misspecified linear regression model. **Matrix structure:** The FIM captures the local curvature of the log-likelihood surface around the current parameter estimate, with brighter colors indicating higher information content. **Diagonal dominance:** The diagonal elements (F_{00} , F_{11} , F_{22}) show the strongest information content, indicating that each parameter contributes independently to the model’s predictive power. **Off-diagonal structure:** The off-diagonal elements reveal parameter interactions and potential redundancy in the model specification. **Optimization implications:** This FIM structure guides natural gradient descent by weighting parameter updates according to local curvature, leading to more efficient convergence than standard gradient descent. The matrix is computed empirically from training data, making it adaptive to the specific data distribution and current parameter values. This empirical approach is particularly valuable when the true data-generating process is unknown or when working with complex, non-linear models where analytical FIM computation is intractable.

- **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.

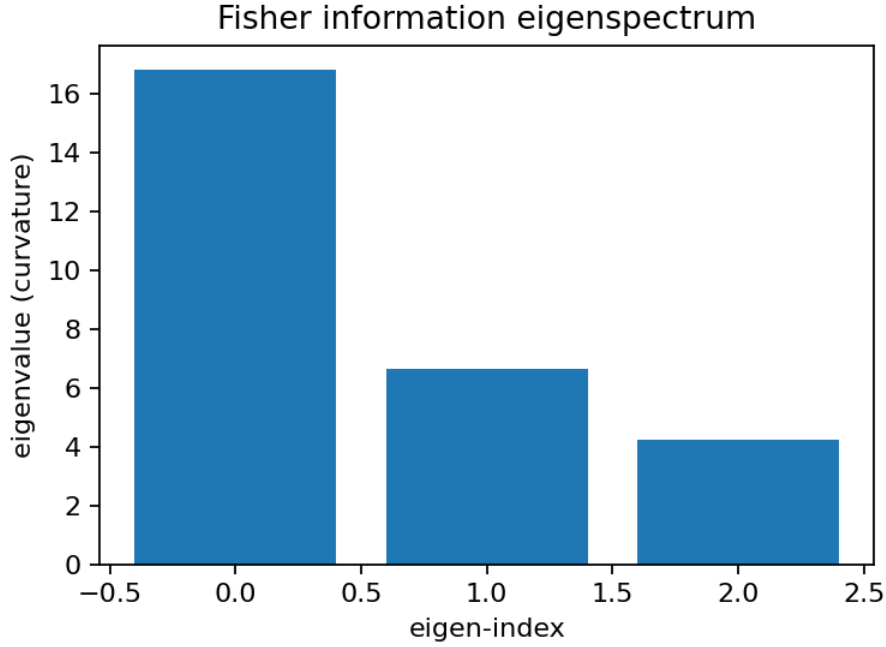


Figure 5: **Fisher Information Matrix eigenspectrum: principal curvature directions.** This bar chart displays the eigenvalue decomposition of the empirical Fisher information matrix from the previous figure, revealing the principal curvature directions of the parameter manifold. **X-axis:** Eigenvalue indices (0, 1, 2) sorted in descending order of magnitude. **Y-axis:** Eigenvalue magnitudes representing the strength of curvature along each principal direction. **Eigenvalue interpretation:** Larger eigenvalues indicate directions of high curvature (tight constraints) where the objective function changes rapidly with parameter changes. Smaller eigenvalues indicate directions of low curvature (loose constraints) where the objective function is relatively flat. **Optimization geometry:** This eigenspectrum reveals the anisotropic nature of the parameter space, explaining why natural gradient descent (which scales updates by the inverse FIM) 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.

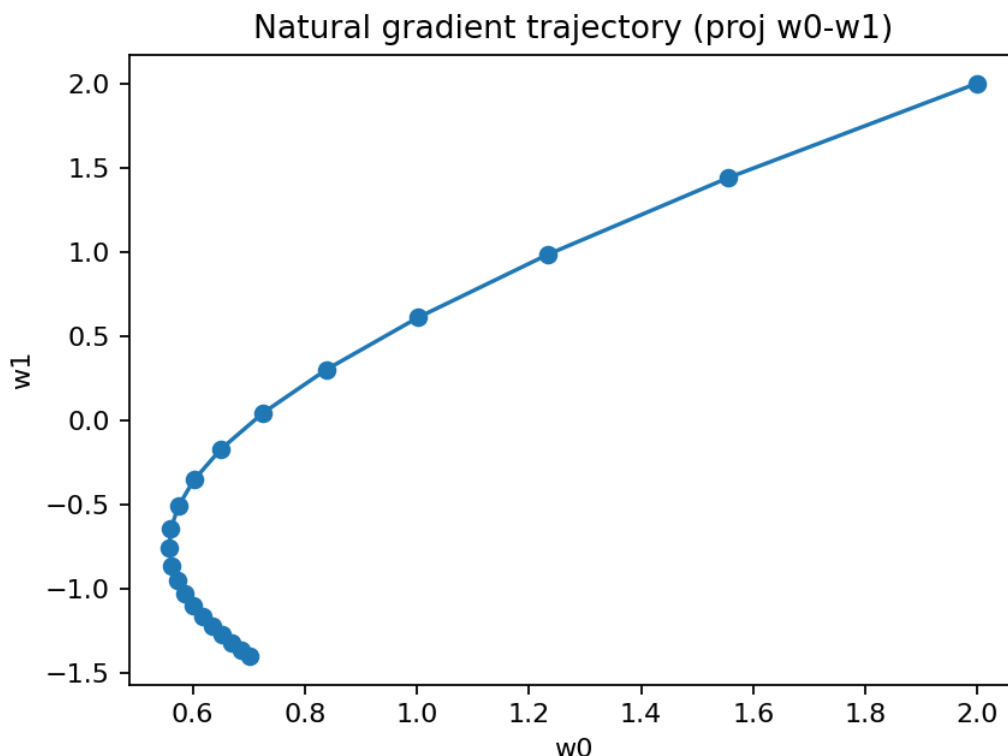


Figure 6: **Natural gradient descent trajectory on a quadratic objective (2D projection)**. This line plot with markers shows the parameter trajectory of natural gradient descent converging to the optimum of a quadratic objective function. **Trajectory**: The blue line with markers traces the parameter 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. **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.

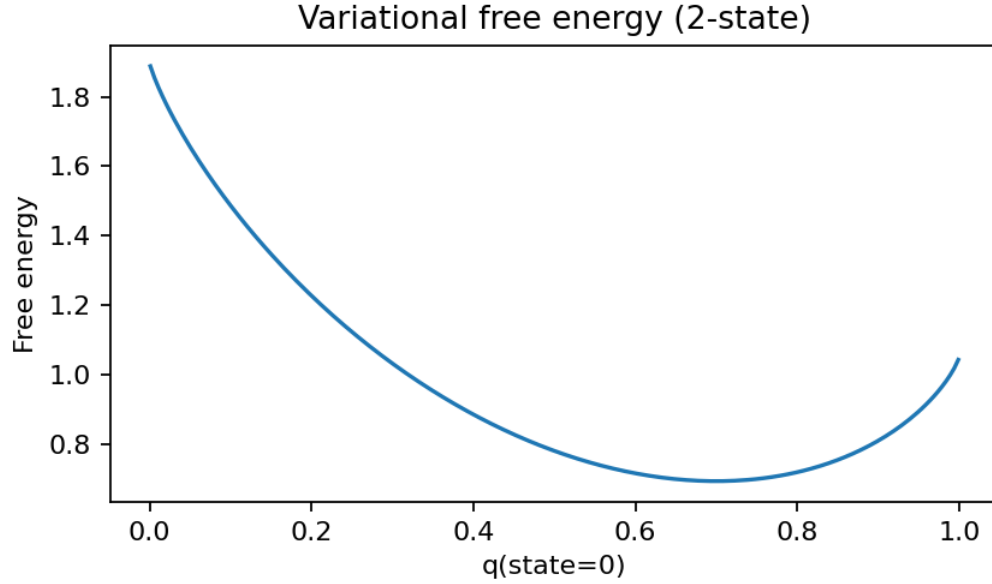


Figure 7: **Variational free energy landscape for a discrete 2-state system.** This curve shows the variational free energy $\mathcal{F} = -\log P(o|s) + \text{KL}[Q(s)||P(s)]$ (see Eq. (??)) as a function of the variational distribution parameter. **X-axis:** Variational parameter controlling the distribution over the two discrete states. **Y-axis:** Free energy value in natural units. **Curve shape:** 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 interpretation:** Minimizing this free energy corresponds to finding the best variational approximation to the true posterior, a fundamental task in Bayesian inference and active inference. The smooth, convex shape of the free energy landscape makes optimization straightforward using standard methods like gradient descent or natural gradient descent. This variational framework provides a principled approach to approximate inference in complex models where exact posterior computation is intractable.

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.