Coherence Dynamics: Bridging Classical and Pure Mathematics

Department of Coherence Dynamics Mathematics

Bi-Topological Dynamics for Certifiable AI: The L'Var Spring in AlphaFold and AlphaGo

Integrating the L'Varian Spring with DeepMind's AlphaFold and AlphaGo

L.E. L'Var (ORCID: 0009-0009-1254-2396)

L'Var Institute of Coherence Dynamics (LICD)

CDM-2025-03 — v1.0 — October 15, 2025

l.e.lvar@the-ap.org — https://the-ap.org

Abstract

DeepMind's AlphaFold and AlphaGo address problems featuring **mixed continuous-discrete dynamics**—optimization/energy descent coupled with hierarchical decisions. The L'Var Spring is a single dynamical engine that runs on two compatible, inequivalent topologies over the same state space: a smooth (Archimedean) topology (τ_∞) for continuous evolution and an ultrametric (p-adic) topology (τ_p) for discrete, hierarchical collapse. The core iteration map L is a strict contraction in both metrics, guaranteeing **dual-certified convergence** to a unique, verifiable fixed point. We propose integrating this framework to replace conventional optimizers and search heuristics. For AlphaFold, this yields a physically-certified folding dynamic, mapping diffusive drift to τ_∞ and decisive conformational "snaps" to τ_p. For AlphaGo, this provides a globally convergent, curvature-aware policy optimizer (τ_∞) coupled with a provably decisive branch-commit mechanism (τ_p), eliminating oscillation. The Spring's categorical structure ensures **compositional safety**: stability and convergence are preserved when certified subsystems are coupled via algebraic pullbacks. This integration elevates these systems from powerful heuristics to provably convergent, geometry-aware architectures.

Keywords: Bi-topological Dynamics, Riemannian Newton Law, Categorical Dynamics, Ultrametric Analysis, AlphaFold, AlphaGo, Certified Safety.

Contents

1 Executive Summary

The L'Var Spring framework offers a strategic upgrade for complex AI systems like AlphaFold and AlphaGo by replacing heuristic stability measures with **mathematically rigorous convergence guarantees**. It is the first unified dynamical system designed to handle the mixed continuous–discrete problems inherent in scientific modeling and strategic planning.

Core Mechanism: Dual-Certified Convergence

he L'Var Spring is a single dynamical engine that runs on two compatible, inequivalent topologies over the same state space:

- τ_∞ **(Smooth/Archimedean)**: Supports continuous evolution and energy descent via an **elastic Riemannian metric** $G(x)$. The continuous limit of the dynamics is the damped Riemannian Newton Law $(G(x)\,\ddot{x} + \Gamma^{\natural}_{G}(x)[\dot{x},\dot{x}] + \zeta(t)\,G(x)\,\dot{x} = -\nabla E(x))$.
- τ_p **(Ultrametric/p-adic)**: Captures discrete, hierarchical "snap" moves via **strict contraction** in an ultrametric distance d_p.

The core iteration map L is a strict contraction in both d_∞ and d_p, meaning every step simultaneously decreases a Lyapunov energy and tightens the solution hierarchy. This property establishes the Spring as a **globally convergent, geometrically preconditioned optimizer** with built-in convergence guarantees.

1.1 AlphaFold: Certified Folding Dynamics

Protein folding is mixed-kinetics: gradual, diffusive adjustments plus sudden, topological "snaps." The Spring maps these directly:

- τ_∞ handles **smooth energy descent** and structure refinement via curvature-aware (Route-B) updates.
- τ_p enforces **ultrametric contraction**, modeling decisive conformational transitions ("perestroikas") that lock stable secondary/tertiary motifs.

The folded native state is the Spring's fixed point ($\mathbf{x}^\star$). **Dual certification** (energy minimum in d_∞ + hierarchical stability in d_p) guarantees terminal convergence without oscillation or failure modes, providing a mathematical certificate of confidence for the predicted structure.

1.2 AlphaGo: Hybrid Optimization That Actually Converges

Strategic game play requires continuous parameter learning (policy/value) and discrete branching (moves). The Spring natively unifies this mixed regime:

- τ_∞ provides **globally convergent, curvature-aware** policy/value updates (Route-B), guiding training along loss surface geodesics.
- τ_p provides **strict ultrametric branch-commit**, pruning to robust subtrees and preventing policy thrash near critical states.

A **dual-certified stopping rule** (calm τ_∞ gradient + fixed ultrametric address) provides a unique terminal strategy, guaranteeing no continuous runaway or discrete dithering.

1.3 Strategic Value: Compositional Safety

The category **L′VarSpring** supplies algebraic pullbacks and related limits. This means systems can be assembled from certified components (networks, search algorithms, controllers) while **preserving the global stability and convergence proofs**. Stability is achieved by architectural design, shifting verification from fragile tuning to formal composition.

Key Performance Indicators and Proposed Research Trajectory

1. **AlphaFold Performance Metrics:** A successful integration should yield statistically significant improvements in key accuracy scores (GDT/lDDT), alongside measurable gains in stability, such as a near-zero reflow rate and strictly monotonic energy decay during refinement.
2. **AlphaGo Performance Metrics:** Validation will be based on improved search decisiveness (quantified by faster Principal Variation stabilization and near-zero branch-flipping events) and superior strategic strength (higher Elo rating at matched computational budgets).
3. **Formal Certification:** The primary deliverable is the formal exclusion of mixed-mode failure states (continuous runaway and discrete oscillation), with certified terminal convergence to a unique fixed point, as guaranteed by the Spring's core theorems.

Proposed Research Trajectory: The immediate research priority is the implementation of a prototype **Route-B Spring refinement loop** and an **ultrametric commit trigger suite** within the existing AlphaFold and AlphaGo architectures. This will serve as the initial validation of the framework's practical applicability and performance benefits.

2 Formal Embedding: Protein Folding as a L'Var Spring

2.1 State, Metric, Ultrametric, and Energy

State space. Let $x = (\phi, \psi, \mathbf{C})$ collect backbone torsions, side-chain rotamers, and Cartesian coords; let Π be a hierarchical contact/correspondence code (e.g., multi-scale contact map or domain tree). Define

$$\mathcal{S} := \underbrace{\mathcal{M}_{\text{geom}}}_{\text{smooth}} \times \underbrace{\mathcal{T}_{\text{hier}}}_{\text{ultrametric}} .$$

Smooth metric (G). On $\mathcal{M}_{\text{geom}}$, set

$$G(x) = W_{\text{geom}}(x) + \lambda_{\text{chem}} W_{\text{chem}}(x) + \lambda_{\text{learn}} W_{\text{NN}}(x),$$

where W_{geom} captures kinematic couplings (torsion/Cartesian Jacobians), W_{chem} penalizes bond/angle/steric violations, and W_{NN} is a learned Fisher-like preconditioner from the model's score network. Assume $G \in C^2$ and uniformly elliptic on level sets of E.

Ultrametric (d_p). Let $\pi : \mathcal{M}_{\text{geom}} \to \Sigma^{\mathbb{N}}$ encode a hierarchical contact signature (e.g., domain→subdomain→motif). Define

$$d_p\big((x, \Pi), (y, \Pi')\big) = \lambda^{-\text{LCP}(\pi(x), \pi(y))} \quad (\lambda > 1).$$

Energy (E). Combine learnable and physical terms:

$$E(x, \Pi) = \underbrace{\alpha E_{\text{NN}}(x)}_{\text{network score}} + \underbrace{\beta E_{\text{phys}}(x)}_{\text{bonds/angles/sterics/electrostatics}} + \underbrace{\gamma E_{\text{hier}}(x, \Pi)}_{\text{hierarchical consistency}} .$$

Here E_{hier} penalizes disagreement between x and its hierarchical code Π (e.g., missing contacts or mis-ordered domain assembly).

2.2 Update Map and Continuous Limit

Route B (prox/accelerated) update on $\mathcal{M}_{\text{geom}}$:

$$y_k = x_k + \beta_k (x_k - x_{k-1}),$$
$$G(y_k)(x_{k+1} - y_k) = -\eta_k \nabla_x E\big(y_k, \Pi_k\big).$$

Ultrametric collapse on $\mathcal{T}_{\text{hier}}$: set Π_{k+1} to the argmin of $E_{\text{hier}}(x_{k+1}, \cdot)$ inside a radius-ρ d_p-ball around Π_k (guarantees strict d_p-contraction for small ρ).

Theorem 2.1 (Bi-topological convergence for folding)**.** *Assume $G \in C^2$ uniformly elliptic, $E \in C^2$ with bi-proper sublevels. With $\beta_k = 1 - \gamma h$ or $1 - {}^{\alpha h}/_{t_k}$ and small ρ, the coupled update yields trajectories whose interpolation solves*

$$G(x)\ddot{x} + \Gamma^{\flat}_G(x)[\dot{x}, \dot{x}] + \zeta(t)G(x)\dot{x} = -\nabla_x E(x, \Pi)$$

while Π performs strict d_p-contractions. Every trajectory converges to a fixed point $(x^\star, \Pi^\star)$; if L is strictly contractive in both metrics, the limit is unique.

Proof sketch. Standard Spring limit for x (Route B) + ultrametric contraction for Π; bi-properness traps the orbit; Banach in each metric gives uniqueness. □

2.3 Falsifiable Predictions for L'Var Spring Folding Dynamics

P1: Barrier crossing as discrete steps. The number of ultrametric collapses equals the number of major domain-assembly events. Measurable via sudden drops in E_{hier} and large d_p-jumps; correlates with formation of native core.

P2: No chattering near near-native states. Because L is contractive in d_p, the hierarchy stabilizes (finite number of code changes) before geometric fine-tuning finishes—observable plateau in Π while x still relaxes.

P3: Reliability bump. On decoys with similar continuous energies but different hierarchies, the Spring picks the hierarchy with minimal E_{hier} and cannot oscillate between them. Metric: lower variance across independent runs.

2.4 Algorithm: The L'Var Spring Folding Engine

Algorithm: Spring Folding Engine

```
Input: initial (x0, Pi0), step h,
       momentum schedule beta_k,
       ultrametric radius rho

for k = 1..K:
  y_k = x_k + beta_k * (x_k - x_{k-1})
  x_{k+1} solves: G(y_k)(x - y_k) = -h * grad_x E(y_k, Pi_k)
  Pi_{k+1} = argmin_{Pi' in B_p(Pi_k, rho)} E_hier(x_{k+1}, Pi')

  if ||x_{k+1} - x_k||_G + d_p(Pi_{k+1}, Pi_k) < epsilon:
    break

return (x_K, Pi_K)
```

3 Formal Embedding: Strategic AI as a L'Var Spring

3.1 State Factorization and Interfaces

Let $\theta \in \Theta$ be network parameters (policy+value), s a game state, and T a search tree rooted at s. Define

$$\mathcal{S} := \underbrace{\Theta}_{\text{smooth}} \times \underbrace{\mathcal{T}(s)}_{\text{hierarchical tree}},$$

with product topology (τ_∞, τ_p).

Metric on Θ. Use $G(\theta) = \lambda_1 I + \lambda_2 \widehat{F}(\theta)$, with $\widehat{F}$ a Fisher-like or curvature proxy (learned or accumulated).

Ultrametric on $\mathcal{T}$. Codes are paths from root; d_p from longest common prefix (as with contact hierarchies).

Energy $E(\theta, T)$.

$$E(\theta, T) = \underbrace{\mathbb{E}_{(s,a)}[\ell_{\text{train}}(\theta; s, a)]}_{\text{supervised/RL loss}} + \underbrace{\mu \ell_{\text{consist}}(\theta, T)}_{\text{policy/value–tree consistency}} + \underbrace{\nu \text{WinCert}(T)}_{\text{formalized proof mass}} .$$

Here ℓ_{consist} penalizes disagreement between network posteriors and tree statistics (visit counts, backed-up values). WinCert is a monotone functional increasing when the tree contains certified winning subtrees (e.g., proof-like terminals).

3.2 The Coupled Update: Learning and Planning

Continuous step (Route B on Θ):

$$\theta_{k+1} = \arg\min_{\theta} \ \tfrac{1}{2}|\theta - \tilde{\theta}_k|^2_{G(\tilde{\theta}_k)} + h \partial_\theta E(\tilde{\theta}_k, T_k) \cdot (\theta - \tilde{\theta}_k), \quad \tilde{\theta}_k = \theta_k + \beta_k(\theta_k - \theta_{k-1}).$$

Discrete tree step (strict d_p-contraction):

$$T_{k+1} := \text{LocalExpandAndPrune}(T_k; \theta_{k+1}; \rho),$$

which (i) expands ε-greedy or UCT-like children within a depth/width budget bounded by ρ, (ii) removes branches dominated under θ_{k+1} (and/or keeps a bandit-certified top subset). Properly tuned, this map reduces the set of admissible frontiers in an ultrametric ball and is contractive in d_p.

Theorem 3.1 (Bi-topological convergence for learning+planning)**.** *Under the same regularity on G and bi-properness of E, the coupled map $L : (\theta, T) \mapsto (\theta', T')$ is a strict contraction in d_p on $\mathcal{T}$ and a descent map in d_∞ on Θ. Trajectories converge to a fixed point $(\theta^\star, T^\star)$. If both metrics are strictly contractive (e.g., η within a trust region; ρ small enough), the limit is unique.*

Interpretation. $T^\star$ stabilizes (no flipping between top branches); $\theta^\star$ is a stationary point for the preconditioned training objective consistent with the stabilized tree.

3.3 Measurable Payoffs for L'Var Spring Planning

G1: Anti-thrash guarantee. On tactical puzzles with two near-equal lines, standard MCTS may oscillate; the Springized planner must monotonically shrink the frontier (ultrametric contraction), yielding strictly fewer policy reversals per move.

G2: Better sample efficiency. With $G(\theta)$ as curvature proxy, match or exceed Elo with fewer gradient steps. Metric: steps-to-plateau vs baseline optimizers.

G3: Certified terminality. With WinCert included, the planner cannot loop: a formal stopping certificate is reached; measure failure rate of "no-move indecision" vs baseline.

4 Safety and Certification: Dual Guarantees as Proofs

4.1 Operational Certificates of Convergence

Energy certificate. $E\big((\theta_k, T_k)\big)$ is strictly decreasing off $\text{Fix}(L)$ and bounded below $\Rightarrow$ convergent energy. Empirically log E_k and verify monotone decay.

Discrete stabilization certificate. The number of ultrametric code changes is finite (bounded by the initial code radius divided by the contraction modulus). Log "hierarchy switches" until zero.

Unique limit certificate. If Lipschitz moduli < 1 in both metrics, the fixed point is unique. Bound Lip_∞ from curvature and step; bound Lip_p from ρ.

4.2 Theorem: Exclusion of Mixed-Mode Failure

Theorem 4.1 (No Mixed-Mode Failure)**.** *No trajectory can (i) diverge in τ_∞ or (ii) chatter in τ_p. Thus, neither continuous runaway nor discrete oscillation is possible; termination is guaranteed.*

Proof sketch. Bi-proper sublevels are precompact in both topologies; the energy descent excludes cycles in τ_∞; strict d_p-contraction excludes discrete oscillations. □

5 Experimental and Falsification Plan

5.1 Protocol 1: AlphaFold Integration

Datasets: CASP decoys; small proteins with known domain hierarchies.

Arms: (i) baseline refinement (Adam/L-BFGS), (ii) Springized refinement (Route B + ultrametric collapse).

Metrics: RMSD@k, TM-score, number of hierarchy switches, monotonicity of E_{hier}, run-to-run variance.

Predictions: Spring reduces hierarchy switches to a finite early burst; fewer failures to reach the native basin; improved stability across seeds.

5.2 Protocol 2: AlphaGo Integration

Tasks: Fixed-budget puzzles; mid-game scenarios with competing tactical lines.

Arms: standard MCTS + SGD vs Springized (prox-Riemann + ultrametric prune).

Metrics: branch flip count per move, frontier size over time, steps-to-Elo plateau, certified stops rate.

Predictions: Fewer flip events, smaller stabilized frontier, same/w better Elo at reduced gradient budget.

6 Implementation Notes

Choosing G: Start with block-diagonal $G = \mathrm{diag}(G_{\text{torsion}}, G_{\text{cartesian}})$ or $G = \lambda I + \lambda_F \widehat{F}$. Ensure condition number bounds on G along the run (clip eigenvalues).

Ultrametric radius (ρ): Small enough to guarantee contraction but large enough to allow genuine topology changes early; anneal $\rho \downarrow$.

Schedules: Constant γ for exponential damping in strongly convex basins; t^{-1} for Nesterov-like phases.

Stopping: Use the joint criterion $\|\Delta x\|_G + d_p(\Pi_{k+1}, \Pi_k) < \varepsilon$ (folding) or $\|\Delta\theta\|_G + d_p(T_{k+1}, T_k) < \varepsilon$ (planning).

7 Strategic Value and Scientific Impact

- A geodesically preconditioned optimizer that provably cannot "go weird."
- A hierarchy-aware collapse operator that turns indecision into convergence.
- Compositionality: you can couple modules (networks, planners, constraints) via pullbacks and know stability survives the coupling.

References

[1] L.E. L'Var, "The L'Var Spring as a Bi-Topological Elastic Object: Energy Descent, Dual Discretizations, and the Newtonian Limit," (Core manuscript, 2025).

[2] A. M. Robert. *A Course in p-adic Analysis.* Graduate Texts in Mathematics, vol. 198, Springer-Verlag, New York, 2000.

[3] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods. *Mathematical Programming*, 137(1-2):91–129, 2013.

[4] Z. B. et al., "A thorough study of Riemannian Newton's method," *arXiv:2506.09297*.

[5] L. Wang et al., "Learning Ultrametric Trees for Optimal Transport Regression," in *AAAI*, 2024.

[6] ANR Project, "Supervised Ultrametric Learning."