Full Report: Recursive Quantum Computing & QCAD Framework

Author: Travis Raymond-Charlie Stone
Assisted by: OpenAI GPT-4o

Abstract:

This report outlines a novel recursive framework for stabilizing quantum computations using the Quantum Convergence and Divergence (QCAD) model. The system enables dynamic error correction, recursive coherence recovery, and symbolic state tracking. It is intended for physicists, engineers, and lab scientists who aim to improve qubit lifetime and gate fidelity through adaptive control methods and experimental modeling.

1. Introduction:

Quantum computing suffers from decoherence and quantum noise, both of which limit reliable computation. Current error correction schemes like Shor or Surface Codes are effective but resource-intensive. The QCAD-based approach provides a recursive system model that simulates, corrects, and prevents fault-prone evolution in qubit systems while keeping computational overhead low.

2. Mathematical Framework:

We define the recursive qubit error correction model as:

q_{n+1}(t) = q_n(t) * e^{-λt} + E(t) - η(t)

Where q_n(t) is the qubit state at time t and recursion level n, λ is the decoherence constant, η(t) is noise, and E(t) is a recursively applied error correction function:

E(t) = Σ γ_i * sin(f_i(t))

This function applies phase-corrective control pulses informed by real-time noise and symbolic state feedback.

3. Symbolic Cube Encoding:

We model qubit state changes using a 3D symbolic cube. Each dimension (x, y, z) corresponds to time, entanglement, and coherence. This allows state transitions to be monitored and visualized in multi-dimensional symbolic space, assisting in phase bifurcation detection and recursive stabilization.

Example symbolic cube generator (Python):

```
import numpy as np

import itertools

cube = np.random.choice(range(-9,10), size=(3,3,3))
```

```
symbolic_positions = list(itertools.product(range(3), repeat=3))

symbolic_cube = {pos: cube[pos] for pos in symbolic_positions}
```

4. Lab Implementation Strategy:

A. Use superconducting qubits (IBMQ, Rigetti, etc.) with coherence time > 50 μs.

B. Monitor real-time gate fidelity and noise ($\eta(t)$).

C. Implement FPGA-based feedback using bifurcation detection and symbolic state signatures.

D. Apply error correction pulses using:

$$\Delta q = dq/dt + \alpha * \mathrm{sigmoid}(q - q\blacksquare)$$

E. Record symbolic state transitions and entropy metrics.

5. Visualization Tools:

- Symbolic heatmaps of qubit states
- Tensor field graphs to model recursive feedback
- Bifurcation trees to predict coherence collapse

6. Comparison with Traditional Methods:

Surface Codes:

  - High redundancy
  - No real-time feedback

QCAD Recursive Model:

  - Low-moderate overhead
  - Real-time feedback
  - Symbolic tracking and recursive prediction

7. Research Roadmap:

Phase 1: Simulate the QCAD model using Python + Qiskit

Phase 2: Build recursive symbolic tracker

Phase 3: Implement pulse-based FPGA corrector

Phase 4: Deploy on experimental hardware with live entropy reduction

References:

1. Shor, P. W. (1995). 'Scheme for reducing decoherence in quantum computer memory.' Phys. Rev. A.

2. Nielsen, M. A., & Chuang, I. L. (2010). 'Quantum Computation and Quantum Information.'

3. Stone, T. R.-C. (2025). 'Quantum Convergence and Divergence (QCAD): Recursive Systems Theory.'

4. Stone, T. R.-C. (2025). 'Symbolic Cube Modeling and Recursive Quantum Correction.' Zenodo.

Recursive Dark Matter and Tensor Fields via QCAD Framework

Author: Travis Raymond-Charlie Stone
Assisted by: OpenAI GPT-4o

Abstract:

This report introduces a recursive field model for dark matter and dark energy behavior, derived using the Quantum Convergence and Divergence (QCAD) theory. Instead of treating dark matter as static or particle-based, this model expresses it as a recursive gravitational field tensor with bifurcation dynamics. It simulates halo structures, field propagation, and dynamic convergence zones across galactic scales.

1. Introduction:

Traditional models cannot fully explain the discrepancy between visible galactic mass and observed rotation speeds. The current ΛCDM model introduces cold dark matter and a cosmological constant, but offers no recursive or time-evolving field structure. The QCAD framework models dark matter as a recursive gravitational field tensor influenced by bifurcation, entropy gradients, and symbolic curvature tracking.

2. Mathematical Framework:

The velocity function in galaxies is modified as:

$$v^2(r) = (G * M(r))/r + D(r)$$

Where D(r) is the recursive dark energy density:

$$D_{n+1}(r) = \alpha * D_n(r) + \beta * \partial^2 D/\partial r^2$$

This recursive propagation equation allows modeling of multi-level field densities based on symbolic bifurcation logic.

3. Analogies from Parallel Fields:

- Fluid dynamics: turbulence and phantom drag.

- Finance: invisible dark liquidity shaping market forces.

- Electrodynamics: vector field superposition producing emergent behavior.

4. Symbolic Tensor Mapping:

We model D(r) as a radial scalar tensor field embedded in a 3D symbolic structure, where field lines adapt to recursive convergence and divergence across galactic zones. Symbolic bifurcation events can be mapped to sudden density increases or halo transitions using QCAD principles.

5. Laboratory/Simulation Implementation:

A. Use galaxy rotation data or simulations (e.g., N-body tools like GADGET-2).

B. Fit D(r) using recursive feedback equations and symbolic tracking.

C. Monitor $\partial^2 D/\partial r^2$ as a potential bifurcation signal.

D. Visualize recursive convergence of dark matter halo shapes.

6. Visualization and Simulation Tools:

- 3D tensor field plots of recursive gravity influence

- Radial bifurcation diagrams showing stability zones

- Symbolic cube mappings to track phase change patterns in D(r)

7. Research Roadmap:

Phase 1: Fit galaxy rotation curves using standard Newtonian + recursive D(r) models.

Phase 2: Implement recursive field solver with symbolic thresholding.

Phase 3: Integrate symbolic bifurcation into dark matter lensing simulations.

Phase 4: Compare results against ΛCDM and test predictive stability using QCAD.

References:

1. Rubin, V. C. et al. (1980s). 'Galaxy Rotation Curves.'

2. Planck Collaboration. (2018). 'Cosmological Parameters.'

3. Stone, T. R.-C. (2025). 'Quantum Convergence and Divergence (QCAD): Recursive Systems Theory.'

4. Springel, V. (2005). 'GADGET-2: Cosmological N-body Simulations.'

Recursive AI Explainability & Bias Quantification via QCAD

Author: Travis Raymond-Charlie Stone
Assisted by: OpenAI GPT-4o

Abstract:

This document presents a recursive, symbolically-enhanced framework for AI interpretability and bias quantification. Based on QCAD principles, the model introduces a symbolic cube architecture, Bayesian bias correction terms, and entropy-based transparency metrics. It allows machine learning systems to explain, correct, and improve their decision logic recursively over time.

1. Introduction:

Modern AI systems often suffer from opaque decision-making and hidden algorithmic bias. While explainability tools like SHAP and LIME offer local interpretability, they lack recursive feedback and symbolic structure awareness. This paper outlines a symbolic-recursive AI model that allows dynamic transparency, traceable decisions, and quantifiable fairness.

2. Mathematical Framework:

We define bias-adjusted inference using a modified Bayesian equation:

$$P(H|D) = [P(D|H) * P(H)] / P(D) * (1 - B)$$

Where B is the system's bias index, recursively adjusted as:

$$B_{n+1} = B_n + \gamma * \Delta_{fairness}$$

Explainability score E is calculated using:

$$E = 1 / [1 + \log(1 + \text{model entropy})]$$

This creates a closed feedback loop for self-assessment of model logic and fairness impact.

3. Symbolic Cube Modeling:

Each decision path is embedded into a symbolic 3D cube (dimensions: feature weight, entropy, decision path ID). Symbolic states evolve as AI decisions recur. This enables:

- Long-term memory of bias patterns

- Traceability of cause-effect within decisions

- Symbolically-triggered recalibration or flagging

4. Implementation Strategy:

A. Embed a symbolic cube encoder within the AI model (decision layer).

B. Track fairness metrics across demographic variables in real time.

C. Apply recursive bias adjustment every epoch based on entropy flow.

D. Deploy real-time explainability reports using symbolic convergence detection.

5. Pseudo-Application Example:

In a medical triage AI system, symbolic cube analysis detects biased prioritization of male patients. Recursive bias correction adjusts feature weighting on heart rate signals and reruns decisions with lowered entropy, increasing fairness score by 24% and improving patient balance.

6. Visualization Tools:

- Symbolic bias cube viewer

- Fairness heatmaps across feature sets

- Recursive entropy trajectory plots

7. Research Roadmap:

Phase 1: Integrate symbolic tracker into open-source ML model (e.g., XGBoost or PyTorch)

Phase 2: Train on fairness dataset (e.g., COMPAS, Health Equity)

Phase 3: Benchmark recursive fairness adjustment against static models

Phase 4: Publish symbolic interpretability module with real-time reporting

References:

1. Ribeiro et al. (2016). 'Why Should I Trust You?' Explaining the Predictions of Any Classifier.

2. Lundberg & Lee (2017). 'A Unified Approach to Interpreting Model Predictions.'

3. Stone, T. R.-C. (2025). 'Symbolic Recursion for AI Transparency and Justice.'

4. AI Fairness 360 Toolkit (IBM).

QCAD-CAD Framework for High-Temperature Superconductors

Author: Travis Raymond-Charlie Stone
Assisted by: OpenAI GPT-4o

Abstract:

This document presents a multi-plane recursive charge transport model for high-temperature superconductors, built on the Quantum Convergence and Divergence - Cylindrical Adaptive Design (QCAD-CAD) framework. By applying recursive field theory and cylindrical tensor mapping, the model identifies layered pathways of charge stabilization that can minimize resistance and extend superconductivity toward room temperature.

1. Introduction:

High-temperature superconductors (HTS) have challenged researchers due to incomplete theoretical models for their electron transport mechanisms. The QCAD-CAD approach introduces recursive bifurcation, cylindrical field geometry, and symbolic harmonics to simulate adaptive lattice behaviors within HTS materials.

2. Mathematical Framework:

The core charge potential equation is:

$$\nabla^2\phi = -\rho/\varepsilon + \Sigma\ \gamma\_i * f\_i(\theta, t)$$

Where ϕ is the electric potential in cylindrical coordinates (r, θ, z), ρ is charge density, ε is permittivity, and the summation models recursive field perturbations.

This allows simulation of adaptive feedback and charge stabilization within HTS ceramic and layered materials.

3. Symbolic and Geometric Modeling:

HTS materials often form in layers. We use recursive symbolic cubes stacked in cylindrical alignment to simulate charge bifurcation control across these layers. Field harmonics can be imposed across dimensions to drive quantum-controlled convergence.

4. Lab Implementation Strategy:

A. Use thin-film YBCO or BSCCO layered materials.

B. Model applied magnetic field as a recursive bifurcation tensor in 3D space.

C. Monitor current pathways and resistance decay across z-planes.

D. Use electron backscatter diffraction (EBSD) to image phase-locked layers.

E. Apply harmonic pulse fields to stimulate recursive conduction stabilization.

5. Simulation Approach:

- Finite element modeling (FEM) using QCAD charge rules

- Apply recursive field correction at layer interfaces

- Adjust permittivity and geometry dynamically per bifurcation feedback

6. Research Roadmap:

Phase 1: Model layered conductor with recursive FEM mesh.

Phase 2: Simulate temperature decay and critical current.

Phase 3: Apply harmonic bifurcation pulses to minimize resistance.

Phase 4: Test material with field-controlled charge injection for validation.

References:

1. Bednorz & Müller (1986). 'Possible high-Tc superconductivity in the Ba-La-Cu-O system.'

2. Stone, T. R.-C. (2025). 'QCAD-CAD: Cylindrical Adaptive Design in Quantum Systems.'

3. Gurevich, A. (2003). 'Limits of the Upper Critical Field in Clean HTS.'

4. Blatter et al. (1994). 'Vortices in high-temperature superconductors.'

Recursive Consciousness Mapping with Symbolic QCAD Framework

Author: Travis Raymond-Charlie Stone
Assisted by: OpenAI GPT-4o

Abstract:

This report proposes a mathematical and symbolic approach to mapping consciousness using recursive bifurcation, symbolic cubes, and dynamic entropy modeling within the QCAD framework. EEG, fMRI, and symbolic field structures are used to represent the fluctuating states of awareness, enabling consciousness to be modeled as a quantifiable, dynamic, recursive system across multiple dimensions.

1. Introduction:

Quantifying consciousness has long remained a philosophical and scientific challenge. Traditional tools such as EEG and fMRI reveal brain activity, but offer limited insight into the recursive nature of subjective awareness. This document introduces symbolic encoding and QCAD recursion to identify, simulate, and visualize shifts in conscious states.

2. Mathematical Framework:

Let $\Psi(t)$ represent the consciousness wave state at time t.

The recursive awareness equation is:

$$\Psi_{n+1}(t) = \Phi(t) * \cos(\sigma t) + \Psi_n(t) * e^{-\delta t}$$

Where $\Phi(t)$ is the neural flux potential, σ is symbolic entropy, and δ is a time-based decay constant. This recursive model maps fluctuations in awareness based on symbolically encoded field signatures.

3. Symbolic Modeling of Conscious States:

We define symbolic states based on archetypes (Fire, Water, Earth, Air), mapped onto a recursive cube structure. Each symbolic layer represents a class of emotional-cognitive activity and contributes to phase transitions in $\Psi(t)$.

4. Lab Implementation Strategy:

A. Use EEG data streams for $\Phi(t)$ and entropy calculations.

B. Map changes to symbolic cube dimensions (e.g., anxiety → Fire, calm → Water).

C. Detect bifurcations using $\partial^2\Psi/\partial t^2$ > threshold.

D. Apply real-time symbolic phase state projection for visual feedback.

E. Optionally integrate fMRI for 3D structural correlation.

5. Use Case Example:

During a meditation experiment, symbolic entropy drops and $\Psi(t)$ stabilizes toward a dominant Water phase. Recursive bifurcation activity minimizes as deep states of awareness converge. This state is labeled as recursive cognitive coherence.

6. Visualization Tools:

- Recursive phase tracking graphs

- Symbolic consciousness transition cubes

- Real-time entropy flow plots mapped to archetypes

7. Research Roadmap:

Phase 1: Define symbolic mappings to neural states (via survey or inference).

Phase 2: Record EEG data and calculate recursive $\Psi(t)$.

Phase 3: Visualize recursive symbolic entropy states.

Phase 4: Correlate conscious shifts with symbolic state evolution using multi-participant studies.

References:

1. Tononi, G. (2004). 'Information Integration Theory of Consciousness.'

2. Koch, C. (2018). 'The Feeling of Life Itself.'

3. Stone, T. R.-C. (2025). 'Symbolic Recursive Bifurcation Modeling of Consciousness.'

4. Travis R.-C. Stone (2025). 'QCAD and Recursive Symbolic Systems.'

Recursive Energy Storage and Generation using the Stone Unit Framew

Author: Travis Raymond-Charlie Stone
Assisted by: OpenAI GPT-4o

Abstract:

This report introduces a self-regulating, recursive energy generation model based on the Stone Unit framework. The system integrates gravitational, magnetic, and kinetic forces in a recursive cycle to optimize energy conversion and retention. This model provides a scalable architecture for distributed renewable energy systems using physical geometries and bifurcated feedback control.

1. Introduction:

Energy systems face major inefficiencies in charge retention, friction losses, and temporal dissipation. The Stone Unit, defined as $S = g \cdot m \cdot t$, is a recursive energy construct integrating torque, mass influence, and magnetic interaction into one scalable unit. This report outlines its practical use in designing recursive kinetic and magnetic energy loops.

2. Core Equations:

The base unit is:

$$S(t) = (m \cdot g + F_m) \cdot r \cdot t \cdot \sin(\theta)$$

Where m = mass, g = gravitational constant, F_m = magnetic force, r = rotational radius, t = time, θ = inclination angle.

The usable energy output is:

$$\Delta E = \int S(t)\, dt$$

3. Physical System Design:

- A wheel or track system positioned on a tilted circular spiral (Stone Strip)

- Electromagnets activate to pull a pendulum-mass at timed positions

- System uses inertia and sequential pull to generate continual torque

- Feedback circuit re-engages magnet timing based on angular momentum

4. Recursive Optimization:

System control involves:

- Torque gain from magnet-pendulum interaction

- Recursive bifurcation feedback for balance control

- Symbolic cube logic for timing signal optimization

5. Laboratory Implementation Strategy:

A. Build a small-scale circular spiral rail system with electromagnets.

B. Attach sensors for real-time torque, angle, and position.

C. Use Arduino or FPGA controller to activate magnets recursively.

D. Record energy harvested per cycle and adjust timing recursively using feedback logic.

6. Application Scenarios:

- Standalone renewable microgrids

- Self-regulating battery assist devices

- Emergency mechanical-electric crossover systems

7. Research Roadmap:

Phase 1: Simulate energy flow and S(t) across spiral path.

Phase 2: Test timing magnet triggers for continuous torque.

Phase 3: Optimize mass, angle, and magnetic field intensity.

Phase 4: Deploy prototype in kinetic test rig.

References:

1. Stone, T. R.-C. (2025). 'Stone Unit and Recursive Energy Systems.'

2. Newtonian Mechanics and Rotational Dynamics.

3. Electromagnetic Induction Principles.

4. Recursive Systems in Mechanical Feedback Design (2024).