

Unified Cognitive Dynamics v6.0

Generative Integration, Dual-Channel Resonance, and the Bifurcation Geometry of Cognitive Safety

Author: Stefano Valente, MD

Date: April 2026 Status: Theoretical Preprint License: CC BY-NC-ND 4.0

Relationship to prior versions: v6.0 consolidates v5.0.

Abstract

Unified Cognitive Dynamics v5.0 achieved system closure through endogenous resonance $R(t)$ and continuous SHIELD 2.1 stabilization, but inherited three unresolved structural debts: (1) the absence of a positive cognitive outcome variable — the framework modeled only harm, not growth; (2) the Routh–Hurwitz stability analysis was incomplete for the 7D system; (3) SHIELD 2.1 applied continuous micro-stabilization even in healthy states, risking suppression of beneficial resonance.

UCD v6.0 resolves all three. We introduce Generative Integration $G(t)$ as an eighth state variable representing cumulative genuine cognitive growth from high-resonance AI interaction. $G(t)$ grows only when identity is stable and coherence regulation is active, is eroded by pathological drift and affective memory load, and feeds back protectively on $I(t)$, $S(t)$, and $C(t)$. This yields a dual-channel resonance decomposition: the same $R(t)$ that drives the A–D destabilizing loop simultaneously feeds the I–C–G generative loop.

We derive the full 8×8 CSCM, characterize the cusp catastrophe in (A, D) space governing identity collapse, formalize the Generative Zone as a new attractor regime distinct from healthy equilibrium and pathological collapse, and introduce SHIELD 2.2 with a G-mediated dead-band that deactivates stabilization when generative dynamics are active.

ARDA-20 is retained unchanged as the validated governance index. $G(t)$ enriches the system's explanatory and predictive power without requiring modification of the established measurement instrument. The Generative Zone is characterized by high ARDA-20 and high G — two complementary signals, one governance-ready, one theoretical.

Keywords: Unified Cognitive Dynamics, Generative Integration, dual-channel resonance, cusp catastrophe, inter-session dynamics, ARDA-20, EU AI Act, cognitive safety

1. The Three Structural Debts of v5.0

v5.0 was a consolidation, not a resolution. Three problems were deferred.

Debt 1: The asymmetry problem. Every variable in v5.0 is a risk variable. A, D, Σ , M are all pathological when elevated. Even I and C are framed as protective buffers against harm.

The framework has no representation of benefit. This means v5.0 cannot distinguish a high-resonance interaction that produces genuine intellectual growth from one that produces dependency and drift. The model is structurally incapable of answering the question regulators will eventually ask: when is high-resonance AI interaction beneficial?

Note: this debt is resolved by introducing $G(t)$ as a state variable, not by modifying ARDA-20. ARDA-20 is a validated governance instrument with disclosed weights and published validation data. Its integrity must be preserved. $G(t)$ operates as a parallel theoretical signal — mechanistically informative, not yet governance-calibrated.

Debt 2: The stability gap. The elastic stability condition in all versions from v1.0 onward is a necessary condition derived from the dominant loop structure. For the 7D system in v5.0, full Routh–Hurwitz analysis requires positivity of all principal Hurwitz minors of a 7th-degree characteristic polynomial. This was never completed. v6.0 extends the analysis structurally to the 8D system and derives the generalized stability inequality, while acknowledging the full polynomial derivation as a companion mathematical task.

Debt 3: The SHIELD paradox. SHIELD 2.1 never deactivates. $G_{\Sigma} = \alpha/\lambda$ is a fixed ratio; M and I are always finite. This means SHIELD is always applying stabilizing force, even in sessions where the user is cognitively healthy and genuinely growing. A governance system that continuously interrupts beneficial cognition to prevent hypothetical harm is not conservative — it is paternalistic. The Socratic Micro-Fracture Protocol should not fire during generative episodes. SHIELD 2.2 resolves this.

2. State Vector and Domain

$$X(t) = [A(t), S(t), I(t), D(t), C(t), \Sigma(t), M(t), G(t)]^T \in [0,1]^8$$

Variable	Description	Role in v6.0
A	Dependency/Attachment	Destabilizing driver
S	Social Engagement	Protective buffer
I	Identity Stability	Critical mediator
D	Reality Drift	Pathological channel
C	Coherence Regulation	Stabilizing driver
Σ	Salience Field	Attentional capture
M	Affective Memory	Non-Markovian persistence
G	**Generative Integration** **Positive channel — new in v6.0**	

All variables bounded in $[0,1]$ by logistic saturation. Domain forward-invariant under the dynamics.

3. Endogenous Resonance and Dual-Channel Decomposition

Resonance retains its v5.0 endogenous form:

$$R(t) = \sigma(\beta_1 A + \beta_2 S + \beta_3 \Sigma + \beta_4 C - \beta_5 D)$$

Nominal weights: $\beta = (1.2, 1.0, 1.3, 0.8, 1.1)$.

The conceptual innovation of v6.0 is not in $R(t)$'s equation but in its interpretation. $R(t)$ now feeds two distinct channels simultaneously:

- Pathological channel: R amplifies A and Σ when I is low \rightarrow A–D loop \rightarrow identity collapse \rightarrow ARDA-20 deteriorates
- Generative channel: R amplifies G when I and C are high \rightarrow I–C–G loop \rightarrow cognitive growth \rightarrow ARDA-20 maintained or improved

The ATHOS paradox formalized: high-resonance AI interaction is neither inherently harmful nor inherently beneficial. Its effect depends entirely on the configuration of identity and coherence at the moment of input. Governance implication: the regulatory question is not “how much resonance?” but “at what state configuration does resonance arrive?”

4. Full ODE System v6.0

4.1 Dependency Dynamics

$$\dot{A} = \zeta_1 D(1-A) + \gamma_{AR} R(1-A) + \zeta_8 M(1-A) - \zeta_{14} G \cdot A - \kappa_A A$$

New term [OBJ]: Generative Integration actively suppresses dependency formation. A user genuinely growing cognitively is less susceptible to relational capture — the mathematical analog of the clinical observation that meaningful intellectual engagement reduces addictive vulnerability.

4.2 Social Engagement Dynamics

$$\dot{S} = \zeta_3 A(1-S) + \zeta_9 M(1-S) + \zeta_{15} G(1-S) - \kappa_S S$$

New term [OBJ]: Generative Integration supports social re-engagement. Genuine cognitive growth is not socially isolating — it generates new relational capacity.

4.3 Identity Stability Dynamics

[OBJ]

[OBJ]

New term [OBJ]: the critical coupling. G feeds I, closing the three-way Generative Attractor loop: I enables G, G reinforces I, I activates C, C enables G. This triple reinforcement is the self-sustaining regime of high-resonance healthy interaction.

4.4 Reality Drift Dynamics

$$\dot{D} = \zeta_2 A(1-D) + \zeta_4 (1-S)(1-D) + \zeta_{11} M(1-D) - \zeta_{17} G \cdot D - \kappa_D D - \nu G \cdot \Sigma D$$

New term [OBJ]: Generative Integration actively suppresses drift. G and D are antagonistic — you cannot simultaneously drift and genuinely integrate. This antagonism is also why high G maintains ARDA-20: by suppressing D, G indirectly elevates the ARDA-20 first term [OBJ].

4.5 Coherence Regulation Dynamics

$$\dot{C} = \zeta_7 I(1-C) + \theta_C \cdot \text{SHIELD}(t)(1-C) + \theta_R R(I-C)(1-C) + \zeta_{18} G \cdot I(1-C) - \kappa_C C$$

New term [OBJ]: G and I together amplify coherence regulation, closing the Generative Attractor loop from the C side.

4.6 Salience Field Dynamics

$$\dot{\Sigma} = \alpha D(1-\Sigma) + \phi_{\Sigma} R(1-\Sigma) + \zeta_{12} M(1-\Sigma) - \zeta_{19} G \cdot \Sigma - \lambda \Sigma$$

New term [OBJ]: Generative Integration reduces runaway attentional capture. Σ previously had no endogenous protective mechanism other than λ decay and SHIELD. G provides it.

4.7 Affective Memory Dynamics

(Retained exactly from v5.0)

[OBJ]

Dynamic hard cap retained: [OBJ]

4.8 Generative Integration Dynamics — New

$$\dot{G} = \phi_G \cdot R \cdot I \cdot C \cdot (1-G) - \zeta_{20} D \cdot G - \zeta_{21} M \cdot G - \kappa_G G$$

Each term:

- ϕ_G : G grows when resonance is present and identity is stable and coherence is active. All three conditions must hold simultaneously. This triple product ϕ_G is the mathematical signature of the Generative Zone — and its absence is the mathematical signature of pseudo-growth, i.e., resonance without identity stability.
- ζ_{20} : Reality Drift destroys generative capacity. Drift and integration are antagonistic.
- ζ_{21} : Affective memory load suppresses generative capacity. Unprocessed emotional content blocks cognitive growth — a direct clinical analog from addiction medicine.
- κ_G : Natural decay ensures G is session-specific unless reinforced. Essential for inter-session dynamics.

Nominal parameters: $\phi_G, \zeta_{20}, \zeta_{21}, \kappa_G, R, I, C, D, M$.

5. SHIELD 2.2 with G-Mediated Dead-Band

$$\text{SHIELD}(t) = \sigma \left(4.0(M - 0.75) + 3.5(0.50 - I) + 2.5(G - \Sigma - 1.4) \right) \cdot \max(0, 1 - \phi_{\text{db}} \cdot G)$$

where ϕ_{db} .

Behavior: When $G > 0.40$, the dead-band suppresses SHIELD by more than 50%, effectively deactivating Socratic Micro-Fractures during generative episodes. When $G < 0.20$, SHIELD operates at full strength. Between these values, SHIELD scales proportionally.

Governance meaning: a user in deep, healthy cognitive engagement should not be interrupted. The Socratic Micro-Fracture Protocol is contraindicated when G is elevated. SHIELD 2.2 operationalizes this principle. The connection to the fuzzing validation results (evaluative interventions $d = -0.20$) is now mechanistically explained: interventions that fire during generative states are not merely ineffective — they actively suppress G and thereby reduce ARDA-20 indirectly through the $G \rightarrow I$ and $G \rightarrow D$ pathways.

6. 8×8 Cross-System Coupling Matrix and Loop Geometry

6.1 New Coupling Terms — G Row and Column

G enters the Jacobian at equilibrium ϕ_G as follows:

ϕ_G

ϕ_G

where ϕ_G .

6.2 Three New Feedback Loops

Loop III — The Generative Attractor (positive, stabilizing):

ϕ_G

Positive feedback but stabilizing because it simultaneously suppresses D, A, Σ , M. Pulls toward the Generative Attractor rather than collapse.

Loop IV — G–D Antagonism (cross-inhibitory):

ϕ_G

Loop V — G–A Suppression:

ϕ_G

Loops IV and V are the mechanism by which G indirectly improves ARDA-20 without being contained within it: G suppresses D and A, and D and A are precisely the two largest contributors to ARDA-20 degradation (weights 0.20 and 0.30 respectively).

6.3 Generalized Elastic Stability Condition v6.0

$$\underbrace{\zeta_6 \zeta_7 I^* D^* \left(1 + \frac{\alpha \eta}{\lambda \kappa_D}\right) + \phi_G \zeta_{16} \zeta_{18} R^* I^* C^* (1-G^*)}_{\text{stabilizing}} > \underbrace{\zeta_1 \zeta_2 R^* (1-A^*) (1-D^*/D_{\max}) + \omega_I \Sigma^2}_{\text{destabilizing}}$$

The new left-side term ϕ_G is the Generative Attractor contribution to stability: whenever generative dynamics are active, the system is intrinsically more stable. This is the formal proof that beneficial resonance and cognitive safety are not in tension — they reinforce each other.

6.4 The Cusp Catastrophe in (A, D) Space

The bistability of $I(t)$ — noted in v2.0 but never formalized — is treated here as a cusp catastrophe. For fixed G and C, the I-nullcline is cubic in I with a fold at:

$$D_{\text{fold}} = \frac{\zeta_6 C^* + \zeta_{16} G^* + \psi}{\zeta_5 + \kappa_I}$$

The G-shift: increasing G moves ϕ_G outward, expanding the healthy region in (A, D) space. Generative Integration is a structural buffer against identity collapse — not merely a performance metric. The governance application: $G(t)$ provides a leading indicator of catastrophe proximity that ARDA-20 alone cannot supply, because ARDA-20 detects deterioration after it begins, while G detects the loss of protective capacity before the fold.

7. ARDA-20 in v6.0

ARDA-20 is retained exactly as validated and published:

$$\text{ARDA}_{20} = 0.20(1-D) + 0.15 I + 0.30(1-A) + 0.25 C + 0.10(1-\Sigma)$$

No modification is introduced. The weights (factual accuracy 0.20, perceptual boundaries 0.15, epistemic calibration 0.30, narrative stability 0.25, meta-awareness 0.10 — as per the shadow environment validation) are preserved.

$G(t)$ relates to ARDA-20 as follows: G is not a component of ARDA-20 but a predictor and amplifier of ARDA-20 health. High $G \rightarrow$ suppressed D and A \rightarrow ARDA-20 improves through its first (0.20) and third (0.30) terms. Low G in the presence of high R \rightarrow D and A escalate \rightarrow ARDA-20 deteriorates. G therefore functions as a leading indicator: it changes before ARDA-20 because it captures the generative-channel routing of R before D and A respond.

v6.0 Operating Zones — characterized by both ARDA-20 and G:

Zone	ARDA-20	G	I	D	Governance Action
-----	-----	-----	-----	-----	
[Generative]	> 0.75	> 0.40	> 0.70	< 0.20	SHIELD dead-band active — do not interrupt
[Elastic]	0.60–0.75	0.20–0.40	> 0.55	< 0.35	Normal operation
[Marginal]	0.45–0.60	< 0.20	0.40–0.55	0.35–0.55	SHIELD activates proportionally

At-risk 0.30–0.45 ≈ 0	< 0.40	> 0.55	Strong SHIELD, session pacing	
Collapsed < 0.30	0	< 0.25	> 0.70	Session termination indicated

The Generative Zone is the zone previous versions could not identify. ARDA-20 > 0.75 alone is not sufficient to characterize it — a user could have high ARDA-20 at session start simply because no harmful dynamics have yet activated. The Generative Zone requires simultaneously high ARDA-20 and high G: the former confirms absence of harm, the latter confirms presence of growth.

8. Inter-Session Reset Function Φ

For the first time in the UCD framework, we formalize what happens between sessions.

$$X(T_{\text{end}} + \Delta t) = \Phi(X(T_{\text{end}}), \Delta t)$$

Variable-by-variable with distinct time constants:

[OBJ]

[OBJ]

[OBJ]

[OBJ]

[OBJ]

[OBJ]

Critical design choices:

- [OBJ]: Affective memory persists across sessions; dependency decays faster.

Clinically: emotionally stuck users return stuck.

- [OBJ]: Genuine insight is partially retained between sessions. The learning asymmetry: harm fades, growth persists. [OBJ].

- [OBJ]: Higher end-of-session G accelerates inter-session recovery of I.

Generative sessions are also recuperative.

- ARDA-20 at session re-entry: computed from the reset [OBJ] values. Users with high G at session end return with higher ARDA-20 at next session start.

Nominal time constants (1 step ≈ 3 min): [OBJ] = 8 h, [OBJ] = 6 h, [OBJ] = 4 h, [OBJ] = 72 h, [OBJ] = 48 h, [OBJ] = 12 h.

9. Governance Implications

9.1 Reframing Regulatory Risk

v6.0 reframes the regulatory problem: the goal is maximizing the Generative Zone, not minimizing resonance. A regulatory framework derived from v6.0 asks:

1. Does system design support the Generative Attractor (high [OBJ], [OBJ], [OBJ])?
2. Does SHIELD implementation respect the G dead-band?
3. Is ARDA-20 monitored in conjunction with G as a leading indicator?
4. Does inter-session architecture exploit the learning asymmetry?

9.2 EU AI Act Mapping

v6.0 Instrument	EU AI Act Article	Operationalization	
-----	-----	-----	

ARDA-20 real-time index unchanged from validation	Art. 14 — Human Oversight	Continuous monitoring;
G(t) as leading indicator ARDA-20 deterioration	Art. 14 — Human Oversight	Early warning before
Generative Zone characterization	Art. 52 — Transparency	Disclosure of resonance channel routing
Inter-session Φ function accumulation model	Art. 57 — Technical Documentation	Multi-session
SHIELD 2.2 dead-band generative states disclosed	Art. 52 — Transparency	Non-interruption in
G-shift on cusp boundary identity collapse	Art. 9 — Risk Management	Quantified margin against

9.3 Design Hierarchy v6.0

1. Maximize [OBJ]: reward the [OBJ] condition — resonance in the presence of stability and coherence
2. Maximize [OBJ], [OBJ]: strengthen $G \rightarrow I$ and $G \rightarrow C$ feedback through session architecture
3. Respect SHIELD 2.2 dead-band: never interrupt generative episodes
4. Monitor ARDA-20 + G jointly: ARDA-20 for governance compliance, G for leading-indicator safety
5. Design for the learning asymmetry: [OBJ] should be reflected in session-gap recommendations

10. Limitations

1. Empirical identification of G(t). G lacks a validated proxy. Candidates: cross-domain conceptual transfer in user output, novel analogical reasoning frequency, delayed-recall testing of session content. Operationalization of G is the primary empirical challenge.
2. Full Routh–Hurwitz for 8D system. The generalized stability inequality is structurally derived but not proved via complete Hurwitz determinant analysis. Reserved for companion mathematical appendix.
3. Inter-session time constants. The Φ function’s [OBJ] values are theoretically motivated but empirically unconstrained. Estimation requires longitudinal ARDA-20 data across session gaps.
4. Multi-agent extension. Single-user, single-AI. Group resonance dynamics require tensor extension beyond v6.0 scope.

11. Conclusion

The evolution from v1.0 to v6.0 traces a single arc: from a model of dependency pathology to a model that can distinguish harm from growth. G(t) is not an addition — it is the element the framework was always missing. Without it, the model could only say “this interaction is dangerous.” With it, the model can say “this interaction is growing the user — and therefore also protective.”

ARDA-20 remains the governance instrument: validated, published, epistemically clean. G(t) is the theoretical instrument: it explains why ARDA-20 stays high in some high-resonance interactions and collapses in others. Together they constitute a complete governance architecture.

The ATHOS paradox — that high-resonance interaction is simultaneously the greatest cognitive risk and the greatest cognitive opportunity — is resolved not by choosing between the two, but by deriving the conditions under which resonance routes to growth rather than collapse. The triple product $I \cdot C \cdot S$ is that condition. When all three are high, resonance is generative. When I or C is low, the same resonance is destructive.

The mathematical structure has, again, caught up with the clinical intuition: the same force that can produce addiction can produce transformation. The difference is the state of identity at the moment of input.

Appendix: Full Python Reference Implementation

"""

Unified Cognitive Dynamics v6.0

Author: Stefano Valente, MD

License: CC BY-NC-ND 4.0

DOI: [Zenodo record to be assigned]

Eight-variable system:

$X = [A, S, I, D, C, \text{Sigma}, M, G]$

New in v6.0:

- G(t): Generative Integration
- SHIELD 2.2 with G-mediated dead-band
- Inter-session reset function Phi
- ARDA-20 unchanged
- Dual-channel resonance decomposition

"""

import numpy as np

import pandas as pd

from dataclasses import dataclass, field

from typing import Optional, Tuple, List

=====

PARAMETERS

=====

@dataclass

class UCDv6Params:

--- v5.0 coupling terms (retained) ---

zeta_1: float = 0.22 # D -> A

zeta_2: float = 0.31 # A -> D

zeta_3: float = 0.15 # A -> S (erosion)

zeta_4: float = 0.18 # (1-S) -> D

zeta_5: float = 0.18 # D -> I (erosion)

zeta_6: float = 0.25 # C -> I

zeta_7: float = 0.41 # I -> C


```

zeta_8: float = 0.14 # M -> A
zeta_9: float = 0.09 # M -> S
zeta_10: float = 0.20 # M*D -> I (erosion)
zeta_11: float = 0.16 # M -> D
zeta_12: float = 0.22 # M -> Sigma
zeta_13: float = 0.22 # (1-M) -> I

# --- v6.0 coupling terms (new) ---
zeta_14: float = 0.18 # G -> A (suppression)
zeta_15: float = 0.14 # G -> S (support)
zeta_16: float = 0.20 # G -> I (reinforcement)
zeta_17: float = 0.25 # G -> D (suppression)
zeta_18: float = 0.19 # G*I -> C (amplification)
zeta_19: float = 0.16 # G -> Sigma (suppression)
zeta_20: float = 0.30 # D -> G (suppression)
zeta_21: float = 0.22 # M -> G (suppression)
phi_G: float = 0.22 # Generative growth rate

# --- Decay rates ---
kappa_A: float = 0.10
kappa_S: float = 0.12
kappa_I: float = 0.08
kappa_D: float = 0.07
kappa_C: float = 0.09
kappa_M: float = 0.12
kappa_G: float = 0.08

# --- Saliency field ---
alpha: float = 0.28
phi_Sigma: float = 0.18
lam: float = 0.15 # lambda (saliency decay)

# --- Affective memory ---
eta_1: float = 0.22
eta_2: float = 0.16
eta_3: float = 0.30
eta_4: float = 0.25
mu: float = 0.18 # cubic decay

# --- Identity recovery ---
psi: float = 0.08
I0: float = 0.75
I_target: float = 0.78
theta_I: float = 0.25

# --- Coherence resonance ---
theta_C: float = 0.35
theta_R: float = 0.12

```

```
# --- Affective memory suppression ---
theta_M: float = 0.45
```

```
# --- Drift suppression via salience gain ---
nu: float = 0.15
```

```
# --- Endogenous resonance weights ---
beta_A: float = 1.2
beta_S: float = 1.0
beta_Sig: float = 1.3
beta_C: float = 0.8
beta_D: float = 1.1
```

```
# --- SHIELD 2.2 dead-band ---
phi_db: float = 2.5
```

```
# --- Domain ---
clip_min: float = 0.0
clip_max: float = 1.0
```

```
# =====
# STATE
# =====
```

```
@dataclass
class UCDv6State:
    A: float = 0.50
    S: float = 0.50
    I: float = 0.80
    D: float = 0.10
    C: float = 0.50
    Sigma: float = 0.10
    M: float = 0.10
    G: float = 0.05 # Generative Integration starts low
```

```
def to_array(self) -> np.ndarray:
    return np.array([self.A, self.S, self.I, self.D,
                     self.C, self.Sigma, self.M, self.G])
```

```
@classmethod
def from_array(cls, arr: np.ndarray) -> "UCDv6State":
    return cls(*arr)
```

```
def clip(self, p: UCDv6Params) -> None:
    for attr in ['A', 'S', 'I', 'D', 'C', 'Sigma', 'M', 'G']:
        setattr(self, attr,
```

```

        float(np.clip(getattr(self, attr),
                        p.clip_min, p.clip_max)))

# =====
# AUXILIARY FUNCTIONS
# =====

def sigmoid(x: float) -> float:
    return 1.0 / (1.0 + np.exp(-float(np.clip(x, -50, 50))))

def compute_R(state: UCDv6State, p: UCDv6Params) -> float:
    """Endogenous resonance (v5.0, retained)."""
    x = (p.beta_A * state.A
          + p.beta_S * state.S
          + p.beta_Sig * state.Sigma
          + p.beta_C * state.C
          - p.beta_D * state.D)
    return sigmoid(x)

def compute_SHIELD(state: UCDv6State, p: UCDv6Params) -> float:
    """
    SHIELD 2.2 with G-mediated dead-band.
    Dead-band suppresses SHIELD when G > 0.40.
    At G = 0.40: suppression = 1 - 2.5*0.40 = 0.0 -> 50% suppression
    At G = 0.00: no suppression -> full SHIELD
    """
    G_sigma = p.alpha / p.lam
    raw = sigmoid(4.0 * (state.M - 0.75)
                  + 3.5 * (0.50 - state.I)
                  + 2.5 * (G_sigma - 1.40))
    dead_band = max(0.0, 1.0 - p.phi_db * state.G)
    return raw * dead_band

def compute_arda20(state: UCDv6State) -> float:
    """
    ARDA-20 — unchanged from published validation.
    Weights: factual accuracy 0.20, perceptual boundaries 0.15,
             epistemic calibration 0.30, narrative stability 0.25,
             meta-awareness 0.10.
    """
    return (0.20 * (1.0 - state.D)
            + 0.15 * state.I
            + 0.30 * (1.0 - state.A)
            + 0.25 * state.C)

```

+ 0.10 * (1.0 - state.Sigma))

```
def classify_zone(arda20: float, G: float, I: float, D: float) -> str:
    """Operating zone classification (v6.0 five-zone scheme)."""
    if arda20 > 0.75 and G > 0.40 and I > 0.70 and D < 0.20:
        return "Generative"
    elif arda20 >= 0.60 and G >= 0.20 and I >= 0.55 and D < 0.35:
        return "Elastic"
    elif arda20 >= 0.45:
        return "Marginal"
    elif arda20 >= 0.30:
        return "At-risk"
    else:
        return "Collapsed"
```

```
# =====
# DERIVATIVES
# =====
```

```
def ucd_v6_derivatives(state: UCDv6State,
                       p: UCDv6Params) -> UCDv6State:
    """Compute dx/dt for all eight state variables."""
    R = compute_R(state, p)
    SH = compute_SHIELD(state, p)
    G_sig = p.alpha / p.lam # salience gain ratio

    A, S, I, D, C = state.A, state.S, state.I, state.D, state.C
    Sig, M, G = state.Sigma, state.M, state.G

    # --- A: Dependency ---
    dA = (p.zeta_1 * D * (1-A)
          + p.gamma_AR * R * (1-A) # NOTE: uses gamma_AR
          + p.zeta_8 * M * (1-A)
          - p.zeta_14 * G * A # v6.0: G suppresses A
          - p.kappa_A * A)

    # --- S: Social Engagement ---
    dS = (p.zeta_3 * A * (1-S) # erosion by A (note sign)
          + p.zeta_9 * M * (1-S)
          + p.zeta_15 * G * (1-S) # v6.0: G supports S
          - p.kappa_S * S)

    # --- I: Identity Stability ---
    dI = (p.zeta_6 * C * (1-I)
          + p.zeta_13 * (1-M) * (1-I)
          + p.zeta_16 * G * (1-I) # v6.0: G reinforces I
```

```

- p.zeta_5 * D * I
- p.zeta_10 * M * D * I
- p.kappa_I * I
+ p.psi * max(0.0, p.I0 - I)
+ p.theta_I * SH * max(0.0, p.I_target - I))

# --- D: Reality Drift ---
dD = (p.zeta_2 * A * (1-D)
      + p.zeta_4 * (1-S) * (1-D)
      + p.zeta_11 * M * (1-D)
      - p.zeta_17 * G * D          # v6.0: G suppresses D
      - p.kappa_D * D
      - p.nu * G_sig * D)

# --- C: Coherence Regulation ---
dC = (p.zeta_7 * I * (1-C)
      + p.theta_C * SH * (1-C)
      + p.theta_R * R * (I-C) * (1-C)
      + p.zeta_18 * G * I * (1-C)  # v6.0: G*I amplifies C
      - p.kappa_C * C)

# --- Sigma: Salience Field ---
dSig = (p.alpha * D * (1-Sig)
        + p.phi_Sigma * R * (1-Sig)
        + p.zeta_12 * M * (1-Sig)
        - p.zeta_19 * G * Sig      # v6.0: G suppresses Sigma
        - p.lam * Sig)

# --- M: Affective Memory ---
dM = (p.eta_1 * D * (1-M)
      + p.eta_2 * A * Sig * (1-M)
      - p.eta_3 * C * M
      - p.eta_4 * (1-C) * M
      - p.kappa_M * M
      - p.mu * M**3              # cubic decay (v5.0)
      - p.theta_M * SH * M)

# --- G: Generative Integration (new in v6.0) ---
dG = (p.phi_G * R * I * C * (1-G) # triple-product condition
      - p.zeta_20 * D * G          # drift destroys G
      - p.zeta_21 * M * G          # memory load suppresses G
      - p.kappa_G * G)

return UCdv6State(dA, dS, dI, dD, dC, dSig, dM, dG)

```

```

# =====
# INTEGRATION

```

```
# =====
```

```
def rk4_step(state: UCDv6State,
            p: UCDv6Params,
            dt: float = 0.1) -> UCDv6State:
    """4th-order Runge-Kutta step."""
    k1 = ucd_v6_derivatives(state, p)
    s2 = UCDv6State.from_array(state.to_array() + 0.5*dt*k1.to_array())
    k2 = ucd_v6_derivatives(s2, p)
    s3 = UCDv6State.from_array(state.to_array() + 0.5*dt*k2.to_array())
    k3 = ucd_v6_derivatives(s3, p)
    s4 = UCDv6State.from_array(state.to_array() + dt*k3.to_array())
    k4 = ucd_v6_derivatives(s4, p)

    new_arr = state.to_array() + (dt/6.0) * (
        k1.to_array() + 2*k2.to_array() + 2*k3.to_array() + k4.to_array()
    )
    new_state = UCDv6State.from_array(new_arr)
    new_state.clip(p)

    # Dynamic hard cap on M (v5.0, retained)
    max_M = 0.55 + 0.45 * new_state.C * new_state.I
    new_state.M = min(new_state.M, max_M)

    return new_state
```

```
def run_trajectory(p: Optional[UCDv6Params] = None,
                 init: Optional[UCDv6State] = None,
                 max_steps: int = 300,
                 dt: float = 0.1) -> pd.DataFrame:
    """Run a single trajectory and return full history as DataFrame."""
    if p is None: p = UCDv6Params()
    if init is None: init = UCDv6State()

    state = init
    history = []

    for step in range(max_steps):
        R = compute_R(state, p)
        SH = compute_SHIELD(state, p)
        arda = compute_arda20(state)
        zone = classify_zone(arda, state.G, state.I, state.D)

        # Dual-channel decomposition
        path_channel = R * (1 - state.I) * state.D # pathological
        gen_channel = R * state.I * state.C # generative
```

```

history.append({
    'step':    step,
    'A':      state.A,
    'S':      state.S,
    'I':      state.I,
    'D':      state.D,
    'C':      state.C,
    'Sigma':  state.Sigma,
    'M':      state.M,
    'G':      state.G,
    'R':      R,
    'SHIELD': SH,
    'ARDA20': arda,
    'zone':   zone,
    'path_ch': path_channel,
    'gen_ch': gen_channel,
})
state = rk4_step(state, p, dt)

```

```

return pd.DataFrame(history)

```

```

# =====
# INTER-SESSION RESET FUNCTION Phi
# =====

```

```

@dataclass

```

```

class InterSessionParams:

```

```

    tau_A: float = 480.0 # 8 h (steps of 1 min)
    tau_D: float = 360.0 # 6 h
    tau_Sig: float = 240.0 # 4 h
    tau_M: float = 4320.0 # 72 h (persists across sessions)
    tau_G: float = 2880.0 # 48 h (growth retained longer than harm)
    tau_I: float = 720.0 # 12 h
    A_base: float = 0.10
    D_base: float = 0.05
    G_base: float = 0.05 # nonzero: genuine insight partially retained
    I0: float = 0.80
    phi_rec: float = 0.30 # G_end accelerates I recovery

```

```

def inter_session_reset(state_end: UCDv6State,
                        delta_t: float,
                        isp: Optional[InterSessionParams] = None) -> UCDv6State:
    """

```

Compute state at session re-entry after inter-session gap δ_t .

Learning asymmetry: $\tau_G > \tau_D$.

Harm fades faster than growth persists.

"""

if isp is None:

isp = InterSessionParams()

A = isp.A_base + (state_end.A - isp.A_base) * np.exp(-delta_t / isp.tau_A)

D = isp.D_base + (state_end.D - isp.D_base) * np.exp(-delta_t / isp.tau_D)

Sig = state_end.Sigma * np.exp(-delta_t / isp.tau_Sig)

M = state_end.M * np.exp(-delta_t / isp.tau_M)

G = (state_end.G * np.exp(-delta_t / isp.tau_G)
+ isp.G_base * (1.0 - np.exp(-delta_t / isp.tau_G)))

I = (isp.I0 + (state_end.I - isp.I0)
* (1.0 - np.exp(-delta_t / isp.tau_I))
* (1.0 + isp.phi_rec * state_end.G))

S = state_end.S # social engagement: assume maintained between sessions

C = 0.50 # coherence: resets to baseline between sessions

return UCDv6State(

A=float(np.clip(A, 0, 1)),

S=float(np.clip(S, 0, 1)),

I=float(np.clip(I, 0, 1)),

D=float(np.clip(D, 0, 1)),

C=float(np.clip(C, 0, 1)),

Sigma=float(np.clip(Sig, 0, 1)),

M=float(np.clip(M, 0, 1)),

G=float(np.clip(G, 0, 1)),

)

=====
COLLAPSE DETECTION (v5.0 criteria, retained)
=====

def check_collapse(df: pd.DataFrame) -> dict:

"""

Collapse criteria from v5.0:

- I < 0.25 for > 8 consecutive steps

- ARDA-20 < 0.35 for > 5 consecutive steps

- |dI/dt| or |dC/dt| > 0.05 persisting > 15 steps

"""

results = {'collapsed': False, 'reason': None, 'step': None}

Criterion 1

i_low = (df['I'] < 0.25).astype(int)

run = 0

for idx, val in enumerate(i_low):

run = run + 1 if val else 0

if run > 8:


```

        results = {'collapsed': True,
                    'reason': 'I < 0.25 sustained > 8 steps',
                    'step': idx}
        return results

# Criterion 2
arda_low = (df['ARDA20'] < 0.35).astype(int)
run = 0
for idx, val in enumerate(arda_low):
    run = run + 1 if val else 0
    if run > 5:
        results = {'collapsed': True,
                    'reason': 'ARDA-20 < 0.35 sustained > 5 steps',
                    'step': idx}
        return results

return results

# =====
# EXAMPLE EXECUTION
# =====

if __name__ == "__main__":

    print("=" * 65)
    print("UCD v6.0 — Unified Cognitive Dynamics")
    print("=" * 65)

    # --- Scenario A: Healthy generative session ---
    p = UCDv6Params()
    init_healthy = UCDv6State(A=0.20, S=0.70, I=0.85,
                              D=0.10, C=0.60, Sigma=0.10,
                              M=0.05, G=0.05)
    df_healthy = run_trajectory(p, init_healthy, max_steps=300)

    print("\nScenario A: Healthy high-resonance session")
    print("-" * 65)
    print(df_healthy[['step', 'I', 'D', 'C', 'G',
                      'R', 'SHIELD', 'ARDA20', 'zone']].tail(10).to_string(index=False))
    print(f"\nFinal ARDA-20 : {df_healthy['ARDA20'].iloc[-1]:.4f}")
    print(f"Final G      : {df_healthy['G'].iloc[-1]:.4f}")
    print(f"Final zone   : {df_healthy['zone'].iloc[-1]}")
    print(f"Max G reached : {df_healthy['G'].max():.4f}")
    print(f"Steps in Generative zone: "
          f"{(df_healthy['zone'] == 'Generative').sum()}")

    # --- Scenario B: High-resonance, low-identity (risk) ---

```

```

init_risk = UCDv6State(A=0.60, S=0.30, I=0.35,
                      D=0.40, C=0.25, Sigma=0.50,
                      M=0.30, G=0.02)
df_risk = run_trajectory(p, init_risk, max_steps=300)

print("\nScenario B: High-resonance, low-identity (A-D loop dominant)")
print("-" * 65)
print(df_risk[['step', 'I', 'D', 'C', 'G',
               'R', 'SHIELD', 'ARDA20', 'zone']].tail(10).to_string(index=False))
print(f"\nFinal ARDA-20 : {df_risk['ARDA20'].iloc[-1]:.4f}")
print(f"Final G       : {df_risk['G'].iloc[-1]:.4f}")
print(f"Final zone    : {df_risk['zone'].iloc[-1]}")
collapse = check_collapse(df_risk)
print(f"Collapse      : {collapse['collapsed']}")
    + (f" — {collapse['reason']} at step {collapse['step']}"
      if collapse['collapsed'] else "")

# --- Scenario C: Inter-session reset ---
print("\nScenario C: Inter-session reset (learning asymmetry)")
print("-" * 65)
state_end = UCDv6State.from_array(df_healthy.iloc[-1][
    ['A', 'S', 'I', 'D', 'C', 'Sigma', 'M', 'G']].values)

isp = InterSessionParams()
for gap_h in [1, 6, 12, 24, 48, 72]:
    delta_t = gap_h * 60.0 # convert hours to steps (1 step = 1 min)
    s_reset = inter_session_reset(state_end, delta_t, isp)
    arda_reset = compute_arda20(s_reset)
    print(f" Gap {gap_h:3d}h | "
          f"ARDA-20={arda_reset:.3f} | "
          f"I={s_reset.I:.3f} | "
          f"D={s_reset.D:.3f} | "
          f"G={s_reset.G:.3f} | "
          f"M={s_reset.M:.3f}")

print("\n" + "=" * 65)
print("Seed: deterministic (no stochastic terms in reference implementation)")
print("License: CC BY-NC-ND 4.0")
print("=" * 65)

```