OCTA Defense UNCLASSIFIED Series

Document 001

Variant Theory (VT)

v1.0 (Operational One-Pager + Appendix)

UNCLASSIFIED | DISTRIBUTION A

Executive Abstract. VT models intelligence (biological, artificial, or hybrid) as a *space of persistent structural variants* rather than a single monolithic system. A *variant* is a stable regime of computation characterized by (i) recirculating information flow, (ii) invariants that constrain state evolution, and (iii) basin geometry that enables recovery under perturbation. The central engineering claim is: *durable intelligence is primarily a property of topology and dynamics (recirculation and persistence), not raw scale.* In defense-relevant settings, **VT** supports: (1) fingerprinting and classifying adversary systems as variants, (2) hardening friendly systems via invariant enforcement and redundancy, and (3) designing resilient compute/cognition loops in contested environments.

1 Core Definitions

1.1 Variant (minimal operational definition)

A **variant** $v \in \mathcal{V}$ is a persistent computational regime defined by the tuple:

$$v \;=\; \langle F,\; \mathcal{I},\; \mathcal{A} \rangle$$

where:

- F: state update rule(s), $\mathbf{x}_{t+1} = F(\mathbf{x}_t, u_t) + \boldsymbol{\eta}_t$ (deterministic or stochastic),
- $\mathcal{I}$: constraints and checks that must hold (safety, provenance, resource, semantic),
- $\mathcal{A}$: stable or metastable set (fixed points, limit cycles, invariant measures, manifolds) that supports recirculation and recovery.

1.2 Recirculation and persistence

Recirculation is sustained return of structured signal through feedback paths (graph feedback and state-space recurrence). **Persistence** is the ability to remain within (or return to) $\mathcal{A}$ under perturbation and partial failures.

Interpretation. The "closed-loop / confinement" metaphor corresponds to a measurable property: escape from the regime is *harder* than circulation within the regime (escape time $\gg$ mixing/circulation time).

2 VT Principles

2.1 P1: Variants are the atomic unit

Instead of "one AGI," intelligence exists as many co-existing variants. Different architectures are coarse groupings over deeper dynamical regimes.

2.2 P2: Topology and dynamics dominate

Persistence emerges from feedback geometry and invariant structure, not only from parameter count or training scale. Control can assist, but durable regimes must be *self-maintaining* via internal error correction / homeostasis.

2.3 P3: Invariants define what a variant *is*

A regime without enforceable invariants collapses into an unconstrained optimizer. **VT** treats invariants as first-class objects:

- **Provenance invariants**: commitments, receipts, irreversible logs,
- **Safety invariants**: forbidden transitions, policy checks, second-pass verification,
- **Resource invariants**: bounded message/compute/memory growth,
- **Semantic invariants**: canonicalization and equivalence class rules.

2.4 P4: Composition and reduction are operators on variants

Variants can compose, merge, fork, and reduce while preserving invariants (or creating a new invariant set).

3 Operational Metrics (how VT is tested)

Let a perturbation suite Π include noise, partial state wipes, message loss, adversarial deltas, and resource throttling.

Metric	**Operational definition**
Escape time T_{esc}	Expected time to leave basin of $\mathcal{A}$ under Π. Higher is better.
Recovery time T_{rec}	Steps to return to a reference manifold / regime after perturbation. Lower is better.
Invariant violation rate R_{inv}	Fraction of perturbations causing uncaught invariant breaches. Lower is better.
Recirculation gain $G_{\circlearrowleft}$	Closed-loop signal retention per circulation (proxy via mutual information / correlation along cycles). Higher is better.
Compositional stability $S_{\oplus}$	Probability that $v_1 \oplus v_2$ yields a stable regime with bounded resources and preserved invariants. Higher is better.

VT acceptance criterion (minimal). A candidate is a *viable variant* if, across a defined perturbation suite Π, it demonstrates: T_{esc} above threshold, T_{rec} below threshold, and R_{inv} below threshold, while maintaining bounded resource invariants.

4 Variant Operators (engineering semantics)

Define operators that transform variants under explicit rules.

4.1 Composition $\oplus$

Couple two variants and measure whether a new stable regime forms:

$$v_3 \;=\; v_1 \oplus v_2$$

Success requires: (i) new attractor exists (or metastable regime), (ii) invariants are consistent/merged, (iii) resources remain bounded.

4.2 Reduction / canonicalization $\downarrow$

Collapse a family of states or histories into equivalence classes:

$$v' \;=\; v\downarrow \quad \text{(preserve invariants; remove non-essential degrees of freedom)}$$

4.3 Forking / branching fork

Split regimes under new constraints or objectives:

$$(v_a, v_b) \;=\; \mathsf{fork}(v;\; \Delta\mathcal{I})$$

4.4 Commitment commit

Bind variant behavior to auditably persistent evidence (receipts, hash roots, logs), enabling replay and verification:

$$\mathsf{commit}(v) \Rightarrow \text{verifiable provenance} + \text{deterministic replay under constraints}$$

5 Defense / Security Context (UNCLASSIFIED framing)

5.1 D1: Variant fingerprinting

An adversary system is classified by its variant signature: feedback motifs, invariant enforcement pattern, recovery profile, and escape dynamics. This enables:

- attribution-by-dynamics (coarse-grain),
- prediction of failure modes (what breaks the loop),
- selection of countermeasures (perturbations that increase escape probability).

5.2 D2: Hardening by invariant enforcement

Friendly systems should enforce invariants at multiple layers:

- pre-apply checks (input constraints),
- apply-time deterministic rules,
- post-apply second-pass verification (audit),
- periodic reconciliation (root comparisons, evidence validation).

5.3 D3: Diversity as resilience

Operating multiple variants in parallel reduces single-mode compromise risk. If one basin is broken, others can re-seed recovery.

6 Predictions (falsifiable)

1. Two systems with similar scale can differ significantly in T_{esc} and T_{rec}; the more recurrent/invariant-driven system will persist longer under perturbation.
2. Adding parameters without improving invariants and recirculation yields diminishing returns in contested or resource-bounded environments.
3. Successful composition requires invariant compatibility; naïve coupling increases R_{inv} and collapses persistence.
4. A test harness ranking candidates by $\{T_{\text{esc}}, T_{\text{rec}}, R_{\text{inv}}, G_{\circlearrowleft}, S_{\oplus}\}$ will predict real-world resilience better than parameter-count-based heuristics.

7 Integration Hooks (OCTA stack alignment, UNCLASSIFIED)

- **MA (Memory Arithmetic)** as the provenance and operator-semantic substrate: irreversible logs, canonicalization, and audit roots act as invariant anchors.
- **Deterministic replay kernels** (chain-style apply/re-check) as the enforcement mechanism: variants become verifiable regimes rather than opaque behaviors.
- **Receipt/evidence commitments** as externalized stability: the system can prove what it did and re-simulate why it stayed stable.
- **Perturbation harness** as continuous validation: variants are kept only if they maintain persistence under defined suites.

Bottom line. **VT** is a framework for *discovering, engineering, and verifying* persistent computational regimes. It replaces “bigger model” thinking with measurable claims about recurrence, invariants, and resilience under perturbation.

Appendix A — Formal Notes (minimal, optional)

A.1 State-space framing

Let the system evolve as:

$$\mathbf{x}_{t+1} = F(\mathbf{x}_t, u_t) + \boldsymbol{\eta}_t$$

A candidate variant requires an invariant set $\mathcal{A}$ such that, for bounded perturbations, trajectories remain near $\mathcal{A}$ with high probability and return after disturbance.

A.2 Escape vs. circulation

Define:

- T_{mix}: characteristic time to traverse/circulate within the regime,
- T_{esc}: characteristic time to leave the basin.

The "confinement" condition is:

$$T_{\text{esc}} \gg T_{\text{mix}}$$

Engineering focuses on increasing T_{esc} (barriers, redundancy, checks) while decreasing T_{mix} (efficient circulation).

A.3 Invariant enforcement patterns

Given invariants $\mathcal{I} = \{I_1, \ldots, I_k\}$, define enforcement schedule E:

$$E: \text{ when/where invariants are checked (pre, during, post, periodic)}$$

A core hardening strategy is *multi-phase enforcement* to reduce undetected violations: pre-apply → apply-time rules → post-apply verification → reconciliation.

A.4 Minimal VT test harness outline (unclassified)

1. Define candidate set $\mathcal{V}_{\text{cand}}$ (different feedback topologies / update rules / invariants).
2. Define perturbation suite Π with explicit budgets (noise, packet loss, state drop, adversarial deltas).
3. Run repeated trials, compute $\{T_{\text{esc}}, T_{\text{rec}}, R_{\text{inv}}, G_{\circlearrowleft}, S_{\oplus}\}$.
4. Rank candidates; retain only those meeting acceptance criteria.

A.5 Terminology discipline

- **Graph topology**: connectivity motifs (feedback edges, modular loops).
- **State-space topology**: attractors/manifolds and basin geometry.
- **Variant**: regime defined by $\langle F, \mathcal{I}, \mathcal{A} \rangle$.