Variant Geometry v1.1 — OCTA Research Edition

Master specification (perfected): metrics, gates, routing, and visual system

OCTA Research

Version 1.1 (Master Perfected) — January 2026

Abstract

Variant Geometry is a computable geometry whose points are *variants*: configurations that satisfy constraints and remain stable under harnessed dynamics. Distances are minimal-cost edit programs between variants, enabling upgrade planning as shortest-path routing in design space. This perfected master v1.1 edition tightens the mathematical foundations (directed metric structure, existence conditions, barrier detours), formalizes OCTA gates and evidence discipline, expands the computable layer (graph sampling, admissible heuristics, multiobjective routing), and delivers a consistent diagram system suitable for spec-grade OCTA documentation.

Contents

1 OCTA Research framing **2**
- 1.1 Why Variant Geometry exists . . . 2
- 1.2 Evidence discipline . . . 2

2 Core primitives **3**
- 2.1 Configuration space . . . 3
- 2.2 Constraints . . . 3
- 2.3 Flow and stability . . . 3
- 2.4 Variants . . . 3

3 Moves, costs, and path lengths **3**
- 3.1 Directed metric structure . . . 3
- 3.2 Geodesic existence (finite case) . . . 4

4 Soft feasibility and risk-adjusted distances **4**
- 4.1 Probabilistic stability . . . 4

5 Gates, admissibility, barriers (OCTA layer) **5**

6 Obstructions: barrier distance and detour curvature **5**

7 Multi-objective (Pareto) routing **6**

8 Computable core (how you actually run it) **6**
- 8.1 Graph construction from a generator . . . 6
- 8.2 Admissible heuristics (practical templates) . . . 6
- 8.3 Algorithms (pseudocode) . . . 7

9 OCTA instantiation layer (templates) **7**
9.1 Artifact classes . . . 7
9.2 Mission relevance $\mathrm{MP_d}$ . . . 8
9.3 Pivot delta vector . . . 8
9.4 Harness report . . . 8
9.5 Regression gate template . . . 8

10 Migration doctrine (visual) **8**

11 Case studies (tightened) **8**

12 Operational checklist (directly implementable) **9**

A Glossary **9**

1 OCTA Research framing

1.1 Why Variant Geometry exists

OCTA treats engineering as navigation through admissible state space:

- **Constraints** encode invariants (safety/security/determinism/consistency).
- **Stability** encodes survivability under harnessed stress (scenario suite Σ).
- **Moves** encode allowed edits (patches, upgrades, rewires).
- **Costs** encode resource/risk/churn/complexity.

A "migration plan" is a shortest admissible path between two variants.

1.2 Evidence discipline

Definition 1.1 (Receipt (abstract))**.** A receipt is a verifiable evidence object $r \in \mathcal{R}$ attached to evaluation or migration supporting claims about invariants and performance.

Definition 1.2 (Evidence gate)**.** An evidence gate is $G_{\text{evi}} : \mathcal{X} \to \{0,1\}$ accepting x iff its harness report includes complete receipts r.

Remark 1.3. Receipts are abstract here. Their cryptographic format (hashes, signatures, timestamps) belongs in separate engineering specs.

2 Core primitives

2.1 Configuration space

Definition 2.1 (Configuration space)**.** Let $\mathcal{X}$ be a set of candidate configurations.

2.2 Constraints

Definition 2.2 (Constraint stack and feasible set)**.** Let $C : \mathcal{X} \to \mathbb{R}^m$. The feasible set is $\mathcal{F} := \{x \in \mathcal{X} : C(x) = 0\}$.

2.3 Flow and stability

Definition 2.3 (Flow / evolution operator)**.** A flow is a family $\Phi_t : \mathcal{X} \to \mathcal{X}$ indexed by $t \geq 0$ or $t \in \mathbb{N}$.

Definition 2.4 (Stability functional)**.** A stability functional is $S : \mathcal{X} \to \mathbb{R}_{\geq 0}$ with $S(x) = 0$ denoting stable and larger values denoting instability.

2.4 Variants

Definition 2.5 (Variants)**.**

$$\mathcal{V} := \{x \in \mathcal{X} : C(x) = 0 \ \wedge \ S(x) = 0\}.$$

Definition 2.6 (Approximate variants)**.** For tolerances $(\delta_C, \delta_S) \in \mathbb{R}^2_{\geq 0}$,

$$\mathcal{V}_{\delta_C,\delta_S} := \{x \in \mathcal{X} : \|C(x)\| \leq \delta_C \ \wedge \ S(x) \leq \delta_S\}.$$

3 Moves, costs, and path lengths

Definition 3.1 (Move set)**.** Let $\mathcal{U}$ be allowed moves. Each $u \in \mathcal{U}$ induces $T_u : \mathcal{X} \to \mathcal{X}$.

Definition 3.2 (Move cost)**.** A move cost is $\kappa : \mathcal{U} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$, written $\kappa(u \mid x)$.

Definition 3.3 (Path and path length)**.** A path $\pi = (u_1, \ldots, u_k)$ induces states $x_0 = a$, $x_j = T_{u_j}(x_{j-1})$. Its length is

$$\mathrm{Len}(\pi; a) := \sum_{j=1}^{k} \kappa(u_j \mid x_{j-1}).$$

Definition 3.4 (Hard distance)**.** For $a, b \in \mathcal{V}$,

$$d(a,b) := \inf_{\pi : a \to b} \mathrm{Len}(\pi; a).$$

3.1 Directed metric structure

Axiom 3.5 (Nonnegativity)**.** $\kappa(u \mid x) \geq 0$.

Axiom 3.6 (Identity move)**.** There exists $e \in \mathcal{U}$ with $T_e(x) = x$ and $\kappa(e \mid x) = 0$.

Theorem 3.7 (Triangle inequality (directed))**.** *For $a, b, c \in \mathcal{V}$, $d(a,c) \leq d(a,b) + d(b,c)$.*

Proof. Concatenate a near-optimal $a \to b$ path and a near-optimal $b \to c$ path and take infima. $\square$

Definition 3.8 (Directed pseudo-metric)**.** A function $d : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ is a directed pseudo-metric if: (i) $d(a, a) = 0$, (ii) $d(a, c) \leq d(a, b) + d(b, c)$.

Proposition 3.9. *Under Nonnegativity and Identity move, the hard distance d is a directed pseudo-metric on $\mathcal{V}$.*

Proof. $d(a, a) = 0$ via identity move. Triangle inequality holds by the theorem above. $\square$

3.2 Geodesic existence (finite case)

Axiom 3.10 (Positive lower bound (finite case))**.** If $\mathcal{X}$ is finite, assume $\exists \alpha > 0$ such that if $T_u(x) \neq x$ then $\kappa(u \mid x) \geq \alpha$.

Theorem 3.11 (Existence of geodesics in finite systems)**.** *If $\mathcal{X}$ is finite and the positive lower bound holds, then whenever $d(a, b) < \infty$ the infimum is attained by some path π.*

Proof. Finitely many path costs exist below any fixed bound due to the positive lower bound; hence a minimum exists among admissible paths. $\square$

4 Soft feasibility and risk-adjusted distances

Definition 4.1 (Violation energy)**.** For $\lambda \geq 0$,

$$\mathrm{E}(x) := \|C(x)\|_2^2 + \lambda S(x).$$

Definition 4.2 (Soft distance)**.** For $\beta \geq 0$, define

$$d_\beta(a, b) := \inf_{\pi : a \to b} \sum_{j=1}^{k} \Big(\kappa(u_j \mid x_{j-1}) + \beta\, \mathrm{E}(x_{j-1}) \Big).$$

4.1 Probabilistic stability

Definition 4.3 (Stability probability)**.** Let Φ_t^ω be randomized. Define

$$\mathsf{Fail}(x) := \{\exists t \in [0, T] : \mathrm{E}(\Phi_t^\omega(x)) > \tau\}, \qquad p_{\mathrm{stab}}(x) := 1 - \Pr[\mathsf{Fail}(x)].$$

Definition 4.4 (Risk-adjusted energy)**.** For $\gamma \geq 0$,

$$\mathrm{E}_{\mathrm{risk}}(x) := \|C(x)\|_2^2 + \lambda S(x) + \gamma(1 - p_{\mathrm{stab}}(x)).$$

Definition 4.5 (Risk-soft distance)**.** Replace E by $\mathrm{E}_{\mathrm{risk}}$ in d_β to obtain $d_{\beta,\gamma}^{\mathrm{risk}}$.

5 Gates, admissibility, barriers (OCTA layer)

Definition 5.1 (Gate)**.** A gate is a predicate $G : \mathcal{X} \to \{0, 1\}$.

Definition 5.2 (Gate stack)**.** Define

$$G_\star(x) = G_{\text{inv}}(x) \wedge G_{\text{reg}}(x) \wedge G_{\text{evi}}(x) \wedge G_{\text{op}}(x).$$

Definition 5.3 (Admissible variants)**.**

$$\mathcal{V}^G := \{x \in \mathcal{V} : \; G(x) = 1\}, \qquad \mathcal{V}^G_{\delta_C,\delta_S} := \{x \in \mathcal{V}_{\delta_C,\delta_S} : \; G(x) = 1\}.$$

Definition 5.4 (Barrier induced by a gate)**.**

$$\mathcal{B}_G := \{x \in \mathcal{X} : \; G(x) = 0\}.$$

Remark 5.5. Gate failure is a first-class barrier. Barrier-aware distances quantify the cost of staying inside admissible design space.

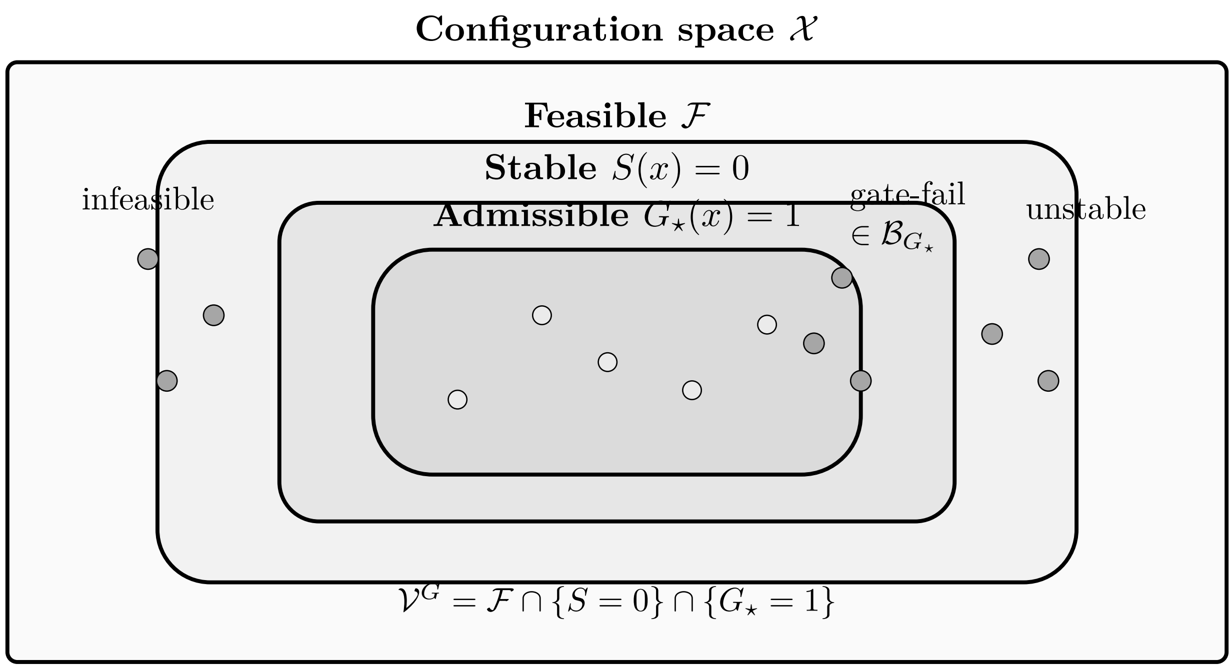

Figure 1: Admissibility layering: feasibility → stability → gate stack.

6 Obstructions: barrier distance and detour curvature

Definition 6.1 (Barrier distance)**.** For barrier set $\mathcal{B} \subset \mathcal{X}$, define $d_{\mathcal{B}}(a, b)$ as the infimum length among paths that avoid $\mathcal{B}$.

Definition 6.2 (Detour factor)**.** If $d(a, b) > 0$,

$$\text{Detour}_{\mathcal{B}}(a, b) := \frac{d_{\mathcal{B}}(a, b)}{d(a, b)}.$$

Proposition 6.3 (Detour bound)**.** *If both distances are finite then* $\text{Detour}_{\mathcal{B}}(a, b) \geq 1$.

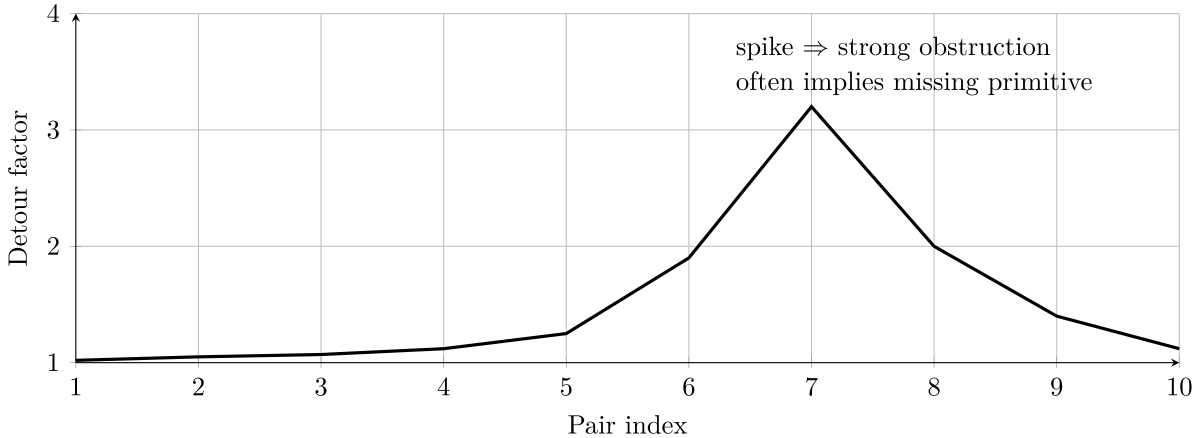

Figure 2: Detour diagnostics: spikes indicate obstruction-heavy regions.

7 Multi-objective (Pareto) routing

Definition 7.1 (Vector cost). Let $\kappa : \mathcal{U} \times \mathcal{X} \to \mathbb{R}^d_{\geq 0}$. Then $\mathrm{Len}(\pi; a) \in \mathbb{R}^d_{\geq 0}$ by summation.

Definition 7.2 (Pareto dominance). $c \prec c'$ if $c_i \leq c'_i$ for all i and $c \neq c'$.

Definition 7.3 (Pareto frontier). $\mathsf{Pareto}(a, b)$ is the set of nondominated path costs from a to b.

Definition 7.4 (Scalarization for search). For $w \in \mathbb{R}^d_{\geq 0}$, define $\kappa_w(u \mid x) = w^\top \kappa(u \mid x)$ and solve for d_w.

8 Computable core (how you actually run it)

8.1 Graph construction from a generator

Definition 8.1 (Move generator). A move generator is a function $\mathsf{Next} : \mathcal{X} \to 2^{\mathcal{X}}$ returning candidate successors of x (with move metadata).

Definition 8.2 (Sampling policy). A sampling policy is a procedure that grows a finite $V \subset \mathcal{X}$ by repeatedly expanding nodes under budget:

$$\text{budget} = (B_{\text{nodes}}, B_{\text{edges}}, B_{\text{time}}).$$

8.2 Admissible heuristics (practical templates)

Definition 8.3 (Heuristic from relaxed costs). Let $\tilde{\kappa} \leq \kappa$ be a relaxation (lower bound). If $h(x)$ is computed as shortest path under $\tilde{\kappa}$, then h is admissible.

Example 8.4 (Common admissible heuristic pattern). If a move changes "edit units" (files, parameters, shards) and each unit has minimum cost $\underline{c}$, then

$$h(x) = \underline{c} \cdot \text{UnitsToGoal}(x)$$

is admissible if UnitsToGoal never overestimates the minimum required units.

8.3 Algorithms (pseudocode)

Listing 1: A* (scalar) with barrier and gate pruning

```
Inputs:
  start a, goal b
  neighbors(x): (y, w(x->y)) edges
  h(x): admissible heuristic
  Gate(x): returns 1 if admissible, else 0
  Barrier(x): returns 1 if forbidden, else 0

A*(a,b):
  if Barrier(a) or Barrier(b): return INF
  open = priority queue by f=g+h
  g[a]=0; parent[a]=None
  push(open,a,f=h(a))

  while open not empty:
    x = pop_min(open)
    if x == b: return g[b], parent
    for (y,wxy) in neighbors(x):
      if Barrier(y): continue
      if Gate(y)==0: continue # strict admissible search
      tentative = g[x] + wxy
      if y not in g or tentative < g[y]:
        g[y]=tentative
        parent[y]=x
        push_or_decrease(open,y,f=tentative+h(y))
  return INF, parent
```

Listing 2: Checkpoint extraction from a computed path

```
Given recovered path nodes x0=a, x1, ..., xk=b:
Checkpoints = [x0]
for i in 1..k:
  if is_rollback_safe(xi) and Gate(xi)==1:
    Checkpoints.append(xi)
Ensure final xk in Checkpoints
Return Checkpoints
```

9 OCTA instantiation layer (templates)

9.1 Artifact classes

Artifact class	Configuration space $\mathcal{X}$ and typical moves $\mathcal{U}$
MA state	state vectors + op-logs; moves: apply delta, reconcile roots, prune, snapshot, rollback.
NUMCHAIN kernel	consensus params + transition rules + caps; moves: rule upgrade, fork-choice tweak, replay format change.

Artifact class	**Configuration space $\mathcal{X}$ and typical moves $\mathcal{U}$**
PERFECT ATTRACTOR router	routing/basin parameters; moves: adjust weights, add/retire basin, change acceptance gates.
ORTHOSPACE engine	basis update rules; moves: change update schedule, stabilization, dimensionality, quantization policy.
SIDECAR infra	gateway/caching/receipts; moves: shard layout, verification mode, SLA policy, replication topology.
Sensor stack	sensors + fusion logic; moves: add modality, recalibrate, reweight, change aggregation.

9.2 Mission relevance $\mathbf{MP_d}$

Definition 9.1 (Mission relevance)**.** Let $\mathrm{Cap}, \mathrm{Prod}, \mathrm{Sus}, \mathrm{Ind} \in [0, 1]$:

$$\mathrm{MP}_d(x) = w_C \mathrm{Cap}(x) + w_P \mathrm{Prod}(x) + w_S \mathrm{Sus}(x) + w_I \mathrm{Ind}(x), \quad \sum w_\cdot = 1.$$

9.3 Pivot delta vector

Definition 9.2 (Pivot delta)**.**

$$\Delta(a \to b) = (\Delta C, \Delta T, \Delta K, \Delta S, \Delta V)$$

= (Cost, Time, Knowledge load, Sustainment burden, Validation strength).

9.4 Harness report

Definition 9.3 (Harness report)**.**

$$\mathcal{H}(x) = (\mathrm{acc},\ \mathrm{lat}_{p50,p95,p99},\ \mathrm{thr},\ \mathrm{viol},\ r).$$

9.5 Regression gate template

Definition 9.4 (Regression gate)**.** Given baseline x_0:

$$G_{\mathrm{reg}}(x) = 1 \iff \mathrm{acc}(x) \geq \mathrm{acc}(x_0) - \Delta_{\mathrm{acc}} \wedge \mathrm{lat}_{p99}(x) \leq \mathrm{lat}_{p99}(x_0)(1 + \Delta_{\mathrm{lat}}) \wedge \mathrm{viol}(x) = 0.$$

10 Migration doctrine (visual)

11 Case studies (tightened)

Example 11.1 (NUMCHAIN migration)**.** Define $\mathcal{X}$ as chain-kernel variants; constraints enforce determinism + consensus; stability is "no stalls under adversarial message patterns." Moves are controlled upgrades (snapshot, rollback, fork-choice tweak, caps). Compute $d^{\mathrm{risk}}_{\beta,\gamma}(a, b)$ and extract rollback-safe checkpoints.

Example 11.2 (SIDECAR detour spike $\Rightarrow$ missing primitive)**.** If $\mathrm{Detour}_{\mathcal{B}_{G_\star}}(a, b)$ is large, admissible paths are obstruction-heavy. Add a new move primitive that creates safe intermediates (e.g., verifiable receipts, trustless verification mode, rollback-safe cache layout), which reduces detour and smooths routing.

12 Operational checklist (directly implementable)

1. Choose artifact class and canonical configuration representation.
2. Implement harness suite Σ producing $\mathcal{H}(x)$ and receipt r.
3. Encode invariants $C(x) = 0$ and stability $S(x) = 0$ from harness traces.
4. Implement gate stack $G_\star$ including evidence gate G_{evi}.
5. Define move primitives and cost model κ (plus risk/energy penalty).
6. Grow a finite move graph under budget and solve routes via Dijkstra/A*.
7. Extract rollback-safe checkpoints and stage the migration plan.
8. Track detour spikes; add primitives to reduce obstructions.

A Glossary

Term	Meaning
Configuration x	Candidate design/state in $\mathcal{X}$
Variant	Feasible ($C = 0$) and stable ($S = 0$) configuration
Admissible variant	Variant that passes a gate (often $G_\star$)
Move u	Allowed edit/transition in $\mathcal{U}$
Cost κ	Nonnegative per-move cost (time/risk/churn/etc.)
Distance d	Minimal path length between variants
Soft distance d_β	Distance penalizing constraint/stability violations along the route
Risk-soft distance	Soft distance penalizing probabilistic failure
Gate stack $G_\star$	Invariant + regression + evidence + ops acceptance gates
Barrier $\mathcal{B}_G$	Gate-failing set of configurations
Detour factor	Inflation due to barriers (obstruction proxy)
Pareto frontier	Nondominated multiobjective costs between variants
Harness report $\mathcal{H}(x)$	Metrics + violations + receipts for x
MP_d	Mission relevance weighting
Pivot delta Δ	Upgrade vector (Cost, Time, Knowledge, Sustainment, Validation)