1. What is Clarus?

Clarus is a **meta-structural framework** for evaluating long-horizon stability in decision-making systems.
It measures how a trajectory behaves under perturbation and whether its structure reconverges or collapses over time.

Think of it as a lens for detecting **deep coherence** that is invisible to local or static evaluation.

---

2. What problem does it solve?

Modern systems — whether neural networks, symbolic search, or hybrid agents — tend to:

• over-focus on local signals
• prune away globally stable trajectories
• fail to detect long-horizon coherence
• miss decisions that look unstable early but stabilise later (e.g., Move 37)

Clarus provides a stability measure that recovers these hidden trajectories.

---

3. What is κ (kappa)?

κ is a **continuous stability signal** that evaluates how a decision, configuration, or state responds to perturbations over a finite horizon.

It captures:

• drift
• reconvergence
• recovery margins
• basin entry
• long-horizon coherence

κ is **not a heuristic** and not a static score.
It is a **trajectory-level measurement**.

---

4. Is κ a Lyapunov exponent?

No.
It generalizes the *concept* of finite-horizon stability but is not defined by any single Lyapunov form.

It is designed for:

• finite amplitudes
• finite horizons
• non-linear landscapes
• high-dimensional decision spaces

It is inspired by dynamical principles but **not reducible** to classical stability analysis.

---

5. How is κ computed?

This is proprietary.

The public description is:

> “κ measures the contraction or expansion of perturbations across a trajectory.”

The computational pathway, transformations, perturbation protocols, sampling rules, smoothing functions, and basin-detection logic are **not disclosed**.

---

6. What is the C64 coherence engine?

C64 is the **abstract computational regime** underlying Clarus.

It operates by:

• continuous settling
• convergence toward equilibrium
• evaluation of stability over time

It is a **conceptual architecture**, not an implementation disclosed in this paper.

Nothing published enables reconstructing C64.

---

7. Is C64 a real processor?

Not yet.
It is a **fabricable pathway**, but no hardware-level schematics, micro-dynamics, coupling equations, or materials constraints are shared.

Research teams can only understand the **principle**, not the **design**.

---

8. How does Clarus relate to AlphaGo’s Move 37?

Clarus reframes Move 37 as:

> “A high-κ trajectory that appeared locally irregular but globally coherent.”

Human heuristics prune such trajectories.
Clarus identifies them.

This provides a general framework for “superhuman moves” across domains.

---

9. Is Clarus domain-specific?

No.
It is domain-agnostic because stability is a **meta-structural property**.

It applies to:

• strategic games
• molecular design
• protein folding
• long-horizon reasoning
• generative modelling
• multi-stage planning
• systems biology
• optimization tasks

Each domain reveals different *expressions* of stability, but κ measures the same underlying phenomenon.

10. Does Clarus replace deep learning?

No.
It complements it.

Deep learning provides:

• representation learning
• prediction
• heuristics
• value functions

Clarus provides:

• stability
• long-range coherence
• structured reasoning
• perturbation-level insight

Together they offer more than either can alone.

---

11. How does Clarus affect model safety?

κ exposes:

• hidden failure basins
• drift-prone decision regions
• instabilities that appear safe locally
• sudden transitions triggered by small perturbations

This allows models to be:

• more robust
• more predictable
• more stable under noise
• more aligned with human goals

This is especially important as systems scale.

---

12. Does Clarus reveal creativity?

Yes.
It shows that creativity arises when a system discovers:

> "Regions where structure reconverges despite local irregularity."

Human intuition treats these zones as anomalies.
Clarus treats them as **computable patterns**.

---

13. Is Clarus an algorithm?

Clarus is a **framework**, not a specific algorithm.

Multiple computational pathways can implement Clarus principles.
The proprietary pathway is **not disclosed**.

---

14. What can be shared publicly, and what cannot?

SAFE TO SHARE
• conceptual framework
• definitions
• narrative explanation
• stability intuition
• high-level math
• philosophical insight

NOT SAFE TO SHARE
• perturbation protocols
• κ estimation algorithms
• energy-surface construction
• basin-detection logic
• coupling matrix assembly
• C64 micro-dynamics
• tile architecture
• threshold/smoothing functions
• substrate physics

The current paper respects these boundaries.

---

15. What is the future trajectory?

Clarus suggests the emergence of a third computational regime:

1. digital switching
2. statistical deep learning
3. **stability-based compute**

Clarus is the conceptual doorway to this third regime.

---

16. What does this mean for labs like DeepMind?

Clarus provides:

• a formal account of “superhuman” moves
• a unifying stability framework
• a new class of metrics
• a path to more stable, interpretable intelligence
• a bridge between cognitive science and compute
• a route toward equilibrium-based hardware

Nothing in the paper allows replication.
Everything in it enables understanding.