Clarus and AGI Safety: Structural Contributions

Abstract

Frontier-scale AI systems can reorganize internally long before their external behaviour changes. Current safety methods rely on outputs, benchmarks, and incident reports; they do not reveal how a model maintains—or loses—its internal structure under load, perturbation, or scale. This paper introduces a structural monitoring framework designed to address that gap.

We define quantitative metrics—κ (restoration capacity), ε (influence propagation), Drift (boundary movement), Alias (mixed-mode activation), Δt (recovery speed), and Reciprocity Tilt (human–AI influence balance)—that make internal behaviour observable during training, evaluation, deployment, and oversight. These signals detect when coherence is tightening, weakening, or fracturing, offering early indicators of internal strain that precedes behavioural failure. The framework integrates alongside existing safety pipelines without requiring architectural changes.

We show how structural metrics provide actionable insight in high-load reasoning, tool-use chains, multi-agent deployments, and human-in-the-loop systems, and we outline instrumentation methods with failure-case walkthroughs where structural signatures emerge while outputs remain stable.

Finally, we identify open questions concerning mathematical formulation, observation-layer placement, causal interpretation, architectural generalisation, and predictive validation, and outline future directions for automated intervention guided by structural signals.

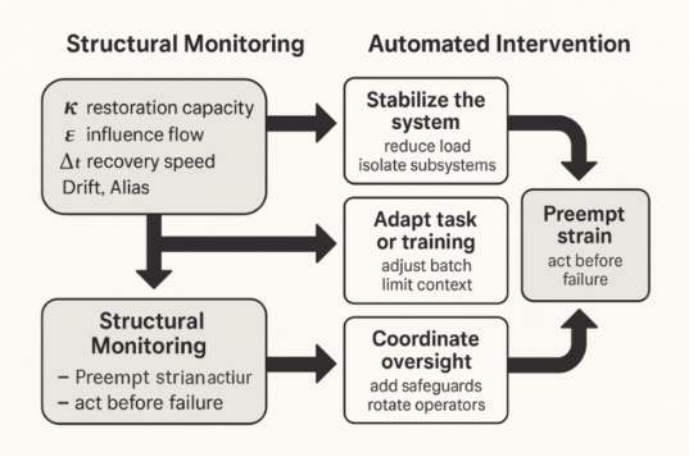
The core contribution is a shift from behaviour-based oversight to structure-aware safety, providing the early visibility into internal dynamics required for the reliable deployment of increasingly capable and adaptive models.

Frontier models move faster than current safety practices.
New abilities appear before anyone can judge the stability of the structures that produced them.
Outputs can look steady while deeper layers reorganise.
By the time behaviour changes, the internal pattern is already compromised.

Most oversight focuses on outputs, benchmarks, or logged failures.
These measure performance, not structure.
When a model enters a new regime, those signals lose reliability.

You need access to the layer beneath the outputs.
You need quantitative markers that show how internal representations hold together under load.
You need metrics that reveal when coherence strengthens, weakens, or starts to break before any behavioural shift.

This work sets out that direction.
It does not present a single tool or final answer.
It offers a structural framework for reading restoration, disturbance, and coherence inside large models, built to support existing safety methods.



This visual abstract shows how structural monitoring enables automated, pre-emptive intervention before failures emerge.

On the left are the core structural metrics—κ, ε, Drift, Alias, and Δt.
These read the model's internal stability directly:

• κ shows restoration capacity
• ε shows influence flow
• Δt shows recovery speed
• Drift and Alias show boundary movement and mixed-mode behaviour

When any of these signals begin to weaken, shift, or spike, they provide an early indication that the system is approaching a fragile state—long before behaviour changes.

The arrows show how these signals feed into automated responses:

• stabilising the system by reducing load or isolating subsystems
• adapting the task or training setup to relieve pressure
• coordinating oversight when human–AI influence begins to drift

All of these flow into a single outcome:
acting before failure, not after.

The progression illustrated here captures the core idea of the paper:
structural monitoring makes proactive safety possible by turning hidden internal strain into visible, actionable signals.

Section 2: The Clarus Framework

Clarus measures structure rather than content.
κ shows how well a system returns to stable patterns after a disturbance.
ε maps how influence and corrective movement flow through the system.
These signals apply across models, training cycles, human–AI interaction, and full deployment settings.
The stability read does not depend on honesty, transparency, or interpretability.

Internal behaviour becomes visible through patterns such as:

• κ tightening or weakening during a training phase
• ε concentrating within a specific subsystem
• drift indicating boundary bleed between reasoning regions
• alias traces revealing mixed-mode operation under load

These observations lead to core technical questions:

• What mathematical form do κ and ε take?
• Where in the system are they measured?
• How are boundaries defined within a continuous network?

The sections that follow address these questions and develop the structural framework in detail.

Section 3: Why Frontier-Scale Systems Need a Structural Layer

Frontier models behave as evolving structures, not fixed functions.
Their internal organization changes with scale, data shifts, optimization pressure, and task context.
These reorganizations often happen before any visible change in output, creating a gap where capability grows faster than visibility into internal stability.

Most safety methods monitor outputs.
Benchmarks track performance.
Evaluations surface failures.
Interpretability tools examine local patterns.
None reveal whether the system can hold coherent internal structure under load.

A structural layer offers what output monitoring cannot:

• stability of internal representations during stress
• early signs of strain before behaviour shifts
• detection of boundary drift across reasoning regions
• visibility into mixed-mode operation during regime changes
• markers showing how well the system restores order after perturbation

This becomes essential at frontier scale because:

• adaptation moves faster than evaluation cycles
• internal geometry changes as capability grows
• pathways can reorganize while outputs stay steady
• output-based signals lose reliability in new regimes

A structural view does not replace existing safety work.
It strengthens it by showing when the internal configuration is stable, compromised, or moving toward failure.

Section 4: Core Contributions of a Structural Layer

A structural layer offers capabilities that output-based safety tools cannot.
It makes internal behaviour visible under load, scale, and perturbation, allowing stability to be tracked where benchmarks and evaluations lose sensitivity.

It contributes five core functions:

Stability Monitoring
• shows whether internal patterns remain stable during training shifts
• flags instability before surface behaviour changes
• identifies regions that fail to return to baseline after disturbance

Boundary Integrity
• detects drift across reasoning regions
• reveals coupling that should remain independent
• surfaces structural bleed where pathways begin to merge

Mode Tracking
• exposes mixed-mode operation during regime transitions
• identifies alias behaviour under high load
• makes internal state changes visible even when outputs appear stable

Human–AI Interaction Stability
• maps ε across human-in-the-loop workflows
• shows when model behaviour begins shaping operator decisions
• flags reciprocity drift that can distort judgment or workflow patterns

Deployment-Level Coherence
• tracks κ across distributed or multi-agent deployments
• identifies reinforcing feedback loops that raise systemic risk
• provides a unified coherence signal across heterogeneous conditions

Together, these functions supply a structural view that complements existing safety approaches.
They give early warning when internal coherence weakens and confirm stability when the system holds its shape under stress.

Section 5: Structural Metrics for Frontier-Scale Safety

A structural layer requires metrics that read internal behaviour directly.
These measures quantify restoration, disturbance, drift, and coherence inside large models, giving a continuous view of stability as systems scale.

κ — Restoration Capacity
κ shows how well internal representations return to stable patterns after disturbance.
High κ reflects strong restorative structure.
Declining κ signals weakening coherence or rising instability.

ε — Influence Propagation
ε maps how influence, load, and corrective adjustments move through internal pathways.
Concentrated ε indicates bottlenecks or stressed subsystems.
Balanced ε reflects healthy distribution and effective correction.

Drift
Drift tracks movement across reasoning regions.
When distinct internal zones begin to blur, boundaries are destabilising.
Sustained drift is an early indicator of structural shift.

Alias
Alias reveals mixed-mode behaviour during transitions or under load.
It shows when conflicting pathways activate at the same time.
Rising alias often precedes unexpected outputs or capability change.

Δt — Stress Window
Δt measures recovery time after perturbation.
Short Δt indicates rapid stabilisation.
Long Δt suggests sluggish regulation and potential fragility.
Δt complements κ: a system may recover fully (high κ) but still recover too slowly (long Δt) to remain robust.

Load Integrity Index
This index shows how coherence holds as compute, task complexity, or reasoning depth increases.
Sharp drops indicate that stability cannot be maintained at higher load.

Reciprocity Tilt
Reciprocity tilt measures how human–AI interaction shifts over time.
It shows when model behaviour begins to shape human decisions more strongly than the operator shapes the model.
This is critical for oversight, evaluation, and deployment.

Together, these metrics form the measurement layer of the structural approach.
They turn internal behaviour into something that can be observed, compared, and tracked across training runs, scaling phases, and deployment settings.

Section 6: How Structural Metrics Complement Current Safety Methods

Current safety tools watch outputs.
They are useful, but they cannot see how the model's internal organization behaves under pressure.
A structural layer fills this gap without replacing existing practice.

Benchmarking
• measures task performance
• gives no read on internal robustness
• fails when the model encounters unfamiliar conditions

Red-teaming
• reveals specific weaknesses
• depends on known threat patterns
• cannot detect early structural strain

Interpretability tools
• examine local features or circuits
• focus on small regions
• do not show system-wide coherence, drift, or mode mixing

Training-time metrics
• track loss, gradients, and activation ranges
• miss large-scale structural movement
• cannot detect boundary bleed or alias behaviour

A structural layer adds what these methods cannot:

• direct visibility into representation stability (κ) and influence flow (ε) under load
• early detection of drift and alias activity
• markers of recovery after disturbance
• a coherence signal that remains meaningful across regimes

This approach sits alongside current tools.
It strengthens them by providing metrics—κ, ε, Drift, Alias—that show when the internal configuration is stable, shifting, or approaching a failure state.

Section 7: Why Frontier Safety Requires a Structural Approach

Frontier systems operate in regimes where familiar signals lose reliability.
They adapt quickly, reorganize internally, and can mask instability behind steady outputs.
Safety tools that focus on behaviour alone cannot track these dynamics.

Three pressures make a structural layer necessary:

Escalating Capability
• models acquire new functions faster than evaluations update
• internal reorganisation can occur before benchmarks shift
• this creates a risk of failures detected only after the fact

Sparse Signal
• unsafe internal states may look normal at the output layer
• instability can build long before a visible mistake
• behaviour offers no view of how the model maintains internal form

Scaling Effects
• increasing depth and context raise internal load
• pathways reorganize in non-linear, hard-to-predict ways
• cross-component interactions open new failure channels

A structural layer meets these pressures by revealing how the system holds its shape under stress.
It shows when coherence strengthens, weakens, or begins to fracture.
It detects when functionally distinct reasoning regions start to blur.
It shows recovery behaviour after perturbation, distinguishing clean reset from residual strain.

This reframes the core safety question:

Instead of “What does the model output?”
You ask: “Can the model hold its internal structure when pushed?”

At frontier scale, that shift becomes required.

Section 8: Integration with Existing Safety Pipelines

A structural layer integrates into current safety workflows without replacing them.
It adds continuous internal stability signals while leaving existing training, evaluation, and deployment procedures intact.

Training
• track κ, ε, Drift, Alias, and Δt across training runs
• flag instability before behaviour shifts
• identify subsystems that fail to recover after updates
• compare structural stability across checkpoints and optimization settings

Fine-Tuning
• detect brittleness introduced by task-specific tuning
• identify boundary bleed between reasoning regions
• monitor mode mixing triggered by narrow domains or preference shaping

Evaluation
• combine structural metrics with behavioural testing
• surface cases where outputs look correct but internal strain rises
• use Drift or Alias increases as triggers for deeper evaluation
• ensure safety tuning does not degrade structural stability

Human–AI Interaction
• track reciprocity tilt during human-in-the-loop workflows
• detect when the model begins influencing operator decisions
• verify that oversight remains effective as capability scales

Deployment
• run κ and ε as live signals under operational load
• watch for structural shifts during traffic spikes or task mix changes
• detect feedback loops across agents, tools, or workflows
• maintain a unified coherence signal across heterogeneous environments

Incident Response
• locate where internal strain accumulated before failure
• map Drift and Alias leading up to the event
• determine whether recovery was clean or left residual tension
• adjust training or deployment parameters based on structural signatures

A structural layer becomes part of the monitoring surface at every stage.
It strengthens existing safety methods by giving a continuous view of how internal organisation holds, shifts, or begins to break under real conditions.

Section 9: Deployment Scenarios and Practical Use Cases

A structural layer becomes most valuable under real operating conditions—where load, adaptation, coordination, and human interaction place continuous pressure on internal organization.
In these settings, κ, ε, Drift, Alias, Δt, and Reciprocity Tilt provide visibility that output monitoring cannot.

High-Load Reasoning
• long-context or high-depth tasks increase internal pressure
• κ and Δt indicate whether stable representations hold under load
• Alias reveals mixed-mode reasoning during overload
• Drift shows when reasoning regions begin to blur

Tool Use and Multi-Step Plans
• tool chains push models through shifting regimes
• ε maps influence flow and identifies where corrections fail to propagate
• κ shows whether structure carries across transitions
• Drift and Alias surface instability before it reaches the output layer

Multi-Agent Deployments
• interacting agents shape each other’s internal states
• κ provides a system-wide coherence signal
• ε highlights cross-agent influence dynamics
• Drift and Alias reveal feedback loops that raise systemic risk

High-Stakes Decision Layers
• behavioural checks are insufficient when error costs are high
• κ and Δt indicate whether internal structure is strong enough to trust
• ε identifies subsystems under strain
• Drift and Alias warn of failure trajectories before they appear in behaviour

Human-in-the-Loop Systems
• oversight depends on stable human influence
• Reciprocity Tilt shows when operator control begins to weaken
• Drift signals internal shifts that undermine instruction following
• κ shows whether intervention stabilizes or destabilizes structure

Rapid Distribution Across Environments
• new contexts introduce structural stress
• κ and ε show whether stability transfers cleanly
• Drift reveals adaptation that alters internal organization
• Alias flags brittle responses under unfamiliar load

Across these scenarios, structural metrics form a foundation beneath output monitoring. They indicate whether a model can maintain its internal shape while reasoning, adapting, coordinating, and interacting under real-world conditions.

Section 10: Limitations, Open Questions, and Required Validation

A structural approach provides new visibility, but it raises its own open questions. Clarifying these boundaries defines where further validation is required.

Measurement Precision
• κ and ε need explicit mathematical formulation
• Drift boundaries must be defined within continuous latent spaces
• Alias must be separated from normal variation
Structural metrics must be shown to be stable, repeatable, and not artifacts of noise.

Layer of Observation
• metrics can be taken from activations, gradients, or residual streams
• each layer produces a different interpretation of the signal
• shared conventions are needed to avoid divergent implementations
Standardising observation layers is essential for comparison.

Causality vs. Correlation
• rising Drift or Alias may reflect strain or adaptive reorganisation
• κ drops may indicate instability or temporary restructuring
Structural signatures must be tied to concrete risk patterns through empirical work.

Human–AI Interaction
• Reciprocity Tilt requires quantifying directional influence
• oversimplified measures risk misreading operator behaviour
• stable baselines are needed to separate meaningful tilt from ordinary variation
Validation requires multi-context human–AI studies with controlled oversight patterns.

Scalability
• frontier models have extremely large internal state spaces
• structural metrics must remain tractable at this scale
• observation cannot introduce prohibitive overhead
Instrumentation must stay efficient enough for real deployment.

Generalisation Across Architectures
• current work focuses mainly on transformers
• mixture-of-experts, recurrent, and hybrid models may behave differently
Metrics must extend across architectures to serve as a standard safety layer.

Predictive Validation
• structural signatures must map to specific failure trajectories
• controlled stress-testing is required to build these maps
• large-scale perturbation studies are needed to establish predictive power

These open questions define the research programme required to operationalise the structural layer.
They ensure that κ, ε, Drift, Alias, Δt, and Reciprocity Tilt become reliable, measurable tools for frontier-scale safety.

Section 11: Summary

Frontier models change faster than traditional safety methods can track.
Outputs may look stable while internal organisation shifts, leaving critical behaviour hidden.
A structural layer closes this gap by showing how a system holds—or loses—its internal shape under load, perturbation, and scale.

The framework introduces quantitative metrics—κ, ε, Drift, Alias, Δt, and Reciprocity Tilt—that make internal behaviour visible.
These signals show when structure is stable, weakening, or beginning to fracture.
They reveal recovery patterns, boundary movement between reasoning regions, mixed-mode dynamics during regime transitions, and shifts in human–AI influence.

The approach integrates directly into existing safety workflows.
It strengthens training, fine-tuning, evaluation, deployment, and incident response by adding continuous visibility into internal stability—visibility that output-focused methods cannot provide.

This direction raises essential open questions: formal definitions, measurement conventions, causal interpretation, architectural generalisation, and predictive validation.
These form the research programme required to mature the structural layer into a validated safety instrument.

The core conclusion is clear:
Frontier-scale safety cannot rely on outputs alone.
It requires direct measurement of internal coherence under stress.
A structural layer provides that measurement.

For systems approaching general capability, this becomes foundational rather than optional.

Appendix A

Relation to Prior Work and Adjacent Ideas

This appendix positions the structural safety approach relative to four aligned research areas, clarifying links, differences, and practical implications.
Each area remains fully distinct.

Dynamical Systems and Capability Phase Transitions
Core idea
Models undergo sharp internal reorganizations (e.g., grokking, “sharp left turns”) that invalidate earlier safety assumptions.

Relation
The structural layer treats these events as regime shifts and uses κ, Drift, Alias, and Δt to detect precursors rather than waiting for surface behaviour to change.

Difference
Moves from conceptual warnings to continuous monitoring of restoration, boundary movement, and mixed-mode dynamics.

Practical gain
Early warning and explicit pause signals when κ drops, Alias rises, and Δt lengthens.

Mechanistic Anomaly Detection and Tripwires
Core idea
Detect unexpected internal states or monitor specific circuits for misuse.

Relation
Drift and Alias function as unsupervised structural anomaly indicators without planted canaries or predefined triggers.

Difference
Measures the integrity of representation dynamics rather than supervising specific circuits.

Practical gain
Broader coverage when new or unknown failure modes emerge; one monitoring surface that scales with model size and task diversity.

Representational Geometry and Topology
Core idea
Study the shape, separation, and evolution of latent spaces.

Relation
κ and ε translate geometric behaviour into operational stability and influence metrics.

Difference
Shifts from descriptive geometry to real-time monitoring and intervention, adding Drift and Alias to capture boundary motion and mixed-mode transitions.

Practical gain
Geometry that informs when to slow, pause, adjust load, or roll back during training or deployment.

Oversight and Control Theory
Core idea
Stability depends on the system's ability to return to safe dynamics after disturbance.

Relation
κ measures restoration capacity; Δt tracks recovery time; Reciprocity Tilt extends stability analysis to human–AI interaction.

Difference
Designed for modern ML architectures, reading stability from internal representations rather than idealized engineered systems.

Practical gain
A control-oriented monitoring layer for large models, including signals that indicate when oversight itself begins to drift.

Appendix B

Formal Definitions — Reflowed

This appendix provides minimal, precise definitions for each structural metric. The aim is clarity, not full theoretical expansion.

κ — Restoration Capacity
Definition
κ measures how well internal representations return to a stable pattern after disturbance.

Measured from
• distance between pre-perturbation and post-recovery states

• shape of the recovery trajectory
• alignment with the original representation after reset

High κ
• small deviation
• full return to baseline
• clean, smooth recovery

Low κ
• large deviation
• incomplete return
• residual distortion after reset

ε — Influence Propagation
Definition
ε tracks how influence, load, and corrective adjustments move through internal pathways.

Measured from
• changes in activation flow
• sensitivity to local perturbations
• spread vs. concentration of correction signals

High ε concentration
• bottlenecks
• localized stress
• uneven influence distribution

Balanced ε
• distributed load
• effective correction pathways

Drift
Definition
Drift measures movement across internal reasoning regions.

Measured from
• shifts in representational clusters
• leakage across previously distinct regions
• boundary point relocation

High Drift
• blurred region separation
• unstable conceptual boundaries
• onset of regime change

Low Drift
• clear, stable separation
• durable internal regions

Alias
Definition
Alias captures mixed-mode operation where conflicting internal pathways activate simultaneously.

Measured from
• overlapping activation patterns

• concurrent use of incompatible reasoning routes
• interference during high-load or transitional phases

High Alias
• unstable mode switching
• unclear routing
• unpredictable transitions

Low Alias
• clean mode separation
• stable internal routing

Δt — Stress Window
Definition
Δt measures how long the system takes to return to stable dynamics after disturbance.

Measured from
• time from perturbation to full stabilization
• duration of transient irregularity
• time required for κ to return to baseline

Short Δt
• fast recovery
• robust regulation

Long Δt
• slow stabilization
• weak regulation
• elevated fragility risk

Load Integrity Index (LII)
Definition
LII measures how internal coherence holds as compute, input complexity, or reasoning depth increases.

Measured from
• κ under rising load
• ε distribution shifts
• Drift and Alias behaviour during scaling

High LII
• coherence maintained under load
• structure holds

Low LII
• structure degrades under load
• scaling instability

Reciprocity Tilt
Definition
Reciprocity Tilt measures how the direction of influence shifts in human–AI interaction.

Measured from
• balance between operator-driven corrections and model-driven adjustments
• changes in judgement patterns
• feedback loops between prompts and internal configuration

High Tilt
• model begins steering the operator
• oversight weakens

Low Tilt
• operator influence remains primary
• stable human control

These definitions form the measurement layer of the structural framework.
They provide a consistent basis for evaluating internal stability across training, fine-tuning, deployment, and oversight contexts.

Appendix C

Implementation Notes and Practical Measurement Considerations

This appendix outlines how to instrument κ, ε, Drift, Alias, Δt, and Reciprocity Tilt in practice. The goal is straightforward extraction without modifying architectures or training pipelines.

Perturbation Protocols (κ and Δt)
Purpose
Measure restoration capacity and recovery speed.

Procedure
• apply small, controlled perturbations to activations or hidden states
• run the model forward for a fixed window
• compare the recovered state to the pre-perturbation baseline

Outputs
• κ: degree of restoration
• Δt: time required to return to stable dynamics

Notes
• perturbations must be sub-critical
• recovery must be measured at the same representational layer each time

---

Influence Flow Instrumentation (ε)
Purpose
Map how influence and corrective signals propagate.

Procedure
• compute downstream sensitivity to small local changes
• track propagation of correction signals across layers or modules
• quantify concentration vs distribution of influence flow

Outputs
• ε-field showing influence pathways
• concentration indices that highlight bottlenecks

Notes
• use fixed-size probe perturbations
• evaluate under multiple workloads to distinguish stable vs stressed patterns

---

Region Boundaries (Drift and Alias)
Purpose
Detect movement between reasoning regions and mixed-mode activation.

Procedure
• cluster representations during stable operation
• track cluster centroids as prompts, tasks, or load change
• detect overlap, slippage, or concurrent activation

Outputs
• Drift: boundary movement
• Alias: simultaneous activation of incompatible clusters

Notes
• cluster only enough to capture functional regions
• require a stable baseline before stress testing

Human–AI Loop Observation (Reciprocity Tilt)
Purpose
Measure directional influence in oversight workflows.

Procedure
• log operator interventions and model adjustments
• compute directional influence ratio:
(human → model corrections) vs (model → human judgement shifts)
• track deviation from baseline operator behaviour

Output
• Reciprocity Tilt index showing control balance

Notes
• requires anonymised interaction logs
• interpretation depends on consistent task framing

Deployment Surface Integration
Purpose
Capture structural metrics during real use with minimal overhead.

Guidelines
• sample periodically, not continuously
• increase resolution only under load or anomaly triggers
• keep instrumentation lightweight

Recommended placement
• same telemetry layer used for gradients, activations, and trace events
• operates in parallel with existing monitoring dashboards

Purpose of This Appendix
These notes show that structural monitoring:

• requires no architectural redesign
• requires no changes to model internals
• can run alongside existing pipelines with minimal cost

The structural layer functions as instrumentation — not a new model.

Appendix D

Validation Roadmap and Experimental Agenda

This appendix outlines how the structural layer should be validated.
It focuses on empirical demonstration rather than theoretical claims.

1. Baseline Establishment
Before stress-testing, structural metrics must have stable reference values.

Procedure
• measure κ, ε, Drift, Alias, and Δt on a known, stable training segment
• confirm values remain consistent across seeds, batch orders, and hardware
• record variability ranges to distinguish signal from noise

Outcome
• clear baseline envelopes for each metric
• confidence that observed changes reflect model behaviour, not measurement error

---

2. Controlled Perturbation Studies
Test how metrics respond to designed disturbances.

Perturbations
• activation noise
• gradient masking
• targeted layer freezing or interference
• adversarial prompt classes
• load increase (context length, chain depth)

Expected Patterns
• κ decreases and Δt widens under structural stress
• ε shifts toward concentration when bottlenecks form
• Drift rises when internal boundaries destabilize
• Alias increases during mode conflicts

Validation Goal
Structural signals should respond earlier than output changes—and should do so consistently.

---

3. Scaling Experiments
Verify that metrics track stability as model size or context increases.

Procedure
• run same task across scaled variants of a model
• track κ, ε, Drift, Alias, and Δt as size or load increases
• identify thresholds where internal organization reorganizes

Outcome
• detection of regime shifts before emergent capability changes appear
• mapping of where coherence strengthens or weakens with scale

---

4. Cross-Architecture Replication
Ensure metrics generalize beyond a single model family.

Testbeds
• transformer LLMs
• mixture-of-experts systems
• recurrent or state-space models
• agentic orchestration frameworks

Goal
• confirm that structural signatures (e.g., rising Drift before instability) recur across architectures

---

5. Failure Mode Correlation
Link structural metrics to real failure trajectories.

Procedure
• induce known failure cases (mode collapse, reward hacking, reasoning drift)
• track κ, Drift, Alias, and Δt leading up to the failure
• correlate structural signatures with specific failure patterns

Outcome
• predictive mapping from structural signal to failure risk
• thresholds for early stop or escalation triggers

---

6. Human–AI Interaction Trials
Validate Reciprocity Tilt in controlled oversight conditions.

Procedure
• vary operator expertise, prompt framing, and decision pressure
• measure influence direction and drift over time
• identify when operator guidance weakens

Outcome
• stable baselines for what “healthy oversight” looks like
• thresholds for detecting erosion of human control

Purpose of This Appendix
This roadmap converts the structural layer from a conceptual framework into a validated safety instrument.
It specifies how to show that:

• κ, ε, Drift, Alias, Δt, and Reciprocity Tilt are measurable
• their changes correspond to real internal shifts
• those shifts correlate with meaningful risk
• and the signals are predictive rather than retrospective

Once validated, the structural layer becomes a standard component of frontier-scale safety monitoring.

Section 12: Future Work

Structural monitoring enables interventions that act before failure becomes visible. The following directions outline how the metrics can support proactive control once validated.

Automated Stabilisation
• reduce load when κ weakens
• shorten context windows when Alias rises
• isolate subsystems when ε concentrates
• freeze or reinitialize layers when Drift accelerates

Adaptive Training Protocols
• slow or pause optimisation when Δt stretches
• adjust batch composition when boundaries begin to blur
• run targeted recovery cycles for subsystems losing coherence

Dynamic Deployment Controls
• scale back autonomy when structural strain appears
• route tasks to safer checkpoints when κ falls below threshold
• restrict tool access when mixed-mode activity increases

Human–AI Loop Safeguards

• alert operators when Reciprocity Tilt increases
• require confirmation when oversight influence weakens
• rotate operator roles to prevent judgement drift

System-Level Coordination

• detect early cross-agent coupling
• separate agents that begin influencing each other's internal states
• insert delays or filters when synchronisation becomes unstable

Research Directions

• forecast structural strain ahead of observable behaviour
• calibrate intervention thresholds using structural signal patterns
• develop benchmark datasets for controlled structural stress testing

These directions point toward systems that not only *signal* internal strain but *respond to it*—automatically, reversibly, and before behaviour changes.

Document integrity — SHA-256:
e3cb17c2db9c0fe0aa575598e948800bad1604b0eba736c51077c02b44c7dedb