From Checklists to Clusters: A Homeostatic Account of AGI Evaluation

Brett Reynolds *
Humber Polytechnic & University of Toronto

October 20, 2025

Abstract

Contemporary AGI evaluations report multidomain capability profiles, yet they typically assign symmetric weights and rely on snapshot scores. This creates two problems: (i) equal weighting treats all domains as equally important when human intelligence research suggests otherwise, and (ii) snapshot testing can't distinguish durable capabilities from brittle performances that collapse under delay or stress. I argue that general intelligence – in humans and potentially in machines – is better understood as a **homeostatic property cluster**: a set of abilities plus the mechanisms that keep those abilities co-present under perturbation. On this view, AGI evaluation should weight domains by their causal centrality (their contribution to cluster stability) and require evidence of persistence across sessions. I propose two battery-compatible extensions: a **centrality-prior score** that imports CHC-derived weights with transparent sensitivity analysis, and a **Cluster Stability Index family** that separates profile persistence, durable learning, and error correction. These additions preserve multidomain breadth while reducing brittleness and gaming. I close with testable predictions and black-box protocols labs can adopt without architectural access.

1 The Brittleness Problem

Hendrycks et al. (2025) present a comprehensive AGI evaluation framework that moves beyond single benchmarks toward rich multidomain profiles. The approach adapts Cattell–Horn–Carroll (CHC) theory (Carroll, 1993; McGrew, 2009; Schneider & McGrew, 2018) to test AI systems across ten domains: general knowledge, reading and writing, mathematics, reasoning, working memory, long-term memory storage and retrieval, visual processing, auditory processing, and processing speed. Each domain gets 10% weight. Systems are tested once and scored.

This is a major improvement over narrow leaderboards. But it still treats intelligence as a flat checklist. Two issues arise:

1.1 The Equal-Weighting Problem

Suppose an AI scores 90% on processing speed but 0% on long-term memory storage. Under equal weighting, that's 9 points toward a 100-point AGI score. But can a system with zero ability to retain new information across sessions really be called generally intelligent? The "memory at 0%"

*I used ChatGPT 5, Claude Sonnet 4.5, and Gemini Pro 2.5 extensively in drafting and revising the paper. I reviewed, edited, and approved all the material and take full responsibility for the final text and conclusions.
brett.reynolds@humber.ca

case isn't hypothetical – it's where many current models sit, relying instead on massive context windows or external retrieval to fake persistence.

Equal weights treat this as equivalent to scoring 90% on knowledge and 0% on speed. But these failures aren't symmetric. A slow system can still learn and reason over time. A system that can't consolidate anything might be fast and knowledgeable *right now*, but it can't build on experience. It's missing something constitutive.

1.2 The Snapshot Problem

Suppose a model scores 70% on reasoning at time $t = 0$. You close the session, wait 24 hours, and retest on similar problems. The score drops to 30%. Which number goes in the AGI evaluation?

Current frameworks report $t = 0$ performance. But if the capability doesn't persist across sessions – if it requires continuously feeding the model its own context or using scaffolds that don't transfer – then the 70% was measuring something temporary. It looked like reasoning but behaved like caching.

1.3 Why This Matters

Brittle, cached, or contorted capabilities can inflate AGI scores without delivering general intelligence. Consider:

- A model that uses a 200K-token context to "remember" user preferences. Disable the context window and it forgets everything – no durable learning occurred.
- A model that scores well on retrieval tasks when RAG is enabled, but produces hallucinations at high rates when RAG is off. The retrieval score was measuring the database, not the model.
- A model that solves reasoning problems by pattern-matching to training data. Delay the test by 72 hours with no access to prior context, and performance collapses because nothing was retained.

These contortions – workarounds that provide temporary performance boosts without genuine capability – are invisible to snapshot testing with equal weights. Evaluation methods are needed that distinguish real capabilities from brittle approximations.

2 From CHC to Homeostatic Property Clusters

2.1 What CHC Gets Right

CHC theory organizes human cognitive abilities into a three-stratum hierarchy (Schneider & McGrew, 2018). At the bottom: dozens of narrow skills (e.g., associative memory, inductive reasoning). In the middle: ∼16 broad abilities (Fluid Reasoning, Working Memory, Long-term Storage and Retrieval). At the top: a general factor g that loads differentially across abilities.

The hierarchy isn't arbitrary. Factor analysis consistently shows that some abilities load more heavily on g than others. Fluid reasoning typically has high g-loading; processing speed has lower loading. This suggests causal structure: some abilities are more central to general intelligence than others.

CHC also documents positive manifold: abilities tend to co-occur. People strong in fluid reasoning tend to be strong in working memory. The correlations aren't perfect – profiles are jagged – but abilities aren't independent.

2.2 What CHC Doesn't Explain

CHC describes the factorial structure of intelligence but doesn't explain *why* abilities cluster or *what keeps them together*. Why should fluid reasoning and working memory correlate? Why does g exist as a statistical regularity?

One answer: intelligence isn't just a collection of abilities. It's a collection of abilities *plus mechanisms that maintain co-occurrence*. When abilities threaten to come apart (e.g., working memory degradation from stress), homeostatic mechanisms compensate (e.g., strategic allocation of attention, reliance on long-term knowledge).

This is the move from taxonomy to dynamics. Intelligence isn't a list of capacities; it's a stabilized cluster.

2.3 Homeostatic Property Clusters in Philosophy of Science

A homeostatic property cluster (HPC) (Boyd, 1999; Khalidi, 2013) is a set of properties that:

1. Tend to co-occur (correlation)
2. Are kept co-present by causal mechanisms (homeostasis)
3. Admit fuzzy boundaries and multiple realizations (graded membership)

Biological species are a cardinal example. Members share morphological, genetic, and behavioral properties. These properties cluster because of mechanisms like reproduction, shared ancestry, and niche adaptation. Boundaries are fuzzy (ring species, hybrids). Realizations vary (mutations, phenotypic plasticity).

Human intelligence fits this pattern, though with an important complication: HPCs can be nested at multiple levels. Abilities like reasoning, memory, and perception cluster *within* a person, and within-agent mechanisms – neural development, synaptic consolidation, executive control, metacognitive strategies – help keep those abilities co-present under perturbation. The *population* also shows a stable clustering: shared developmental environments, schooling, and cultural tools shape the distribution and co-occurrence of abilities across people and over time.

In what follows I assess cluster membership for AI systems at the *agent level*; population-level processes are treated as background conditions.[1]

2.4 Applying HPCs to AGI Evaluation

If intelligence is an HPC, then evaluating AGI requires testing two things:

1. **Abilities** (the cluster nodes): Can the system reason, remember, perceive, etc.?
2. **Stability mechanisms** (the edges): Do those abilities persist when stressed? Can the system compensate when one ability is degraded?

Current AGI evaluation focuses on (1) and ignores (2). But (2) is what makes the cluster robust. A system might have all the right abilities at $t = 0$ but lack the mechanisms to maintain them under perturbation. That's a capability profile, not general intelligence.

A crucial scope clarification: the unit of evaluation throughout is a single agent (model instance). External supports that operate at the population level in humans – shared developmental

[1] Kind-level regularities supervene on the distribution of agent-level profiles *plus* relatively stable environmental institutions. They are related but not identical; collapsing them obscures what the unit under evaluation (a single model) can sustain on its own.

environments, schooling, cultural transmission – are treated here as environmental background. Their AI analogues (context windows, RAG systems, tool access) are assessed as compensatory versus contorted scaffolds in Section 5, with the key question being whether they extend the agent's capabilities or merely substitute for missing ones.

Two immediate implications arise:

1. Not all abilities contribute equally to cluster integrity. Some nodes are upstream (e.g., long-term storage enables cumulative learning). Some are downstream (e.g., processing speed helps but isn't constitutive). Equal weighting ignores this causal structure.

2. Stability is constitutive of cluster membership. A system that scores 90% on reasoning but can't maintain reasoning performance across delays or distribution shifts exhibits weak stability mechanisms. It has demonstrated one-shot pattern-matching rather than robust reasoning. Systems can show varying degrees of stability – membership is graded – but very low stability (near-zero consolidation, no profile persistence, no error correction) suggests the cluster is barely holding together.

3 Causal Centrality: Why Some Abilities Matter More

3.1 Three Kinds of Centrality

"Centrality" can mean different things. To avoid confusion, I distinguish:

- **Structural centrality (s_i)**: Component i is upstream in a causal graph. Example: long-term storage enables cumulative abilities like domain knowledge and procedural skills. Storage is structurally central.

- **Impact centrality (c_i)**: Degrading component i causes performance collapse across multiple other domains. Example: if fluid reasoning drops by 50%, math, reading comprehension, and planning all suffer. Reasoning has high impact centrality.

- **Stabilizing centrality (h_i)**: Enhancing component i improves stability indices (defined in Section 4). Example: adding robust retrieval reduces hallucinations *and* makes the system more reliable under noise. Retrieval has stabilizing centrality.

These distinctions matter because centrality types can come apart in surprising ways. Processing speed might be structurally downstream (it doesn't enable other abilities, so low s_i) but still have moderate impact centrality (losing speed degrades performance everywhere, so moderate c_i) without contributing much to stability (faster processing doesn't make capabilities more durable, so low h_i). Conversely, working memory might have high values across all three types – it's upstream structurally, its degradation harms multiple domains, and enhancing it improves both performance and stability.

3.2 From Centrality to Weights

If components differ in centrality, why weight them equally? The obvious alternative: weight by centrality.

Here's the challenge. Fully estimating impact centrality c_i and stabilizing centrality h_i requires controlled experiments: degrade component i and measure downstream effects, or enhance component i and measure stability gains. For black-box systems, this is difficult. For systems still under

development, it's premature. For humans, it's largely infeasible – we can't ethically degrade working memory or disable storage to measure causal impacts.

A better approach than equal weights is to use a **centrality prior** – a principled starting point based on existing evidence from human cognition research – and report the result with sensitivity analysis. The prior isn't a claim of ground truth. It's a "here's what I'd expect given human data and theoretical considerations, subject to revision as we gather AI-specific evidence". Because we can't run the ideal degradation experiments on humans, the prior has to rely on correlational CHC g-loadings (which reflect abilities that tend to co-vary with general intelligence) rather than experimentally-derived causal centrality measures. This is a limitation, but it's transparent and improvable as AI-specific perturbation data accumulates.

This approach avoids bootstrapping. I don't use stability to define centrality and then use centrality to define stability. Instead, the centrality prior comes from independent evidence (CHC factor analysis and causal structure), while stability indices (Section 4) provide an independent validation check.

The centrality prior uses two sources of information:

- $\tilde{g}_i$: normalized empirical CHC g-loadings from human factor analysis
- $\tilde{s}_i$: normalized structural priors about causal dependencies between domains

I reserve $\tilde{h}_i$ (stabilizing centrality) for future experimental updates once perturbation studies on AI systems become available. This keeps the framework transparent: each weight can be traced back to either human data or explicit mechanistic assumptions.

The tilded quantities represent normalized versions that sum to 1 across domains:

$$\tilde{g}_i = \frac{g_i}{\sum_j g_j}, \qquad \tilde{s}_i = \frac{s_i}{\sum_j s_j}, \qquad \tilde{h}_i = \frac{h_i}{\sum_j h_j}$$

This normalization ensures that when we combine them into weights, the result remains a proper probability distribution: $\sum_i w_i = 1$ provided the mixing parameters also sum to 1.

3.3 A CHC-Informed Prior

CHC theory provides one concrete source of evidence: decades of factor-analytic work have established which abilities load more heavily on the general factor g in human populations (Carroll, 1993; McGrew, 2009). Abilities with high g-loadings are more central to general intelligence (in humans) than abilities with low loadings. This gives us $\tilde{g}_i$, a normalized version of these empirical loadings.

This can be combined with structural priors $\tilde{s}_i$ based on the CHC hierarchy itself and mechanistic considerations about causal dependencies. For instance, storage and retrieval are clearly upstream – you can't have durable knowledge without the ability to consolidate and access information. Reasoning operates over stored content and benefits from efficient retrieval. Speed modulates efficiency but doesn't generate new capabilities on its own.

With both sources in hand, I define the centrality-prior weights as:

$$w_i^{\text{prior}} = \lambda\, \tilde{g}_i + \mu\, \tilde{s}_i, \qquad \lambda + \mu = 1, \quad \lambda \in [0.6, 0.9] \tag{1}$$

Here, λ controls how much weight to give to empirical g-loadings versus structural priors. The range $[0.6, 0.9]$ is chosen to reflect high but not exclusive confidence in the empirical factor-analytic data: high λ (e.g., 0.8) trusts the human CHC loadings heavily, while low λ (e.g., 0.6) hedges by blending in more structural considerations. Values below 0.6 would give equal or greater weight to

structural priors than to decades of factor analysis, which seems unjustified; values above 0.9 would ignore structural considerations almost entirely. By varying λ across this range and reporting the resulting score intervals, we make the sensitivity to this choice explicit.

3.4 An Illustrative Prior (Defeasible)

To make this concrete, here's one example prior chosen to reflect typical CHC g-loadings and structural bottlenecks (Carroll, 1993; Schneider & McGrew, 2018). These percentages are illustrative – different research teams might make different defensible choices – but they provide a starting point:

- Reasoning (R / Gf): 14%
- Comprehension–Knowledge (Gc, K): 13%
- Working Memory (WM / Gwm): 12%
- Mathematics (Gq): 13%
- Reading/Writing (Grw): 12%
- Visual Processing (V / Gv): 8%
- Auditory Processing (A / Ga): 8%
- Long-term Storage (MS; part of Glr): 7%
- Long-term Retrieval (MR; part of Glr): 7%
- Processing Speed (Gs, S): 6%

Two notes on specific components:

First, CHC's broad factor *Long-term Storage and Retrieval* (Glr) encompasses both encoding and retrieval processes. I separate Storage (MS) and Retrieval (MR) for analytic clarity – this decomposition reflects mechanistic considerations about consolidation versus cue-dependent access rather than CHC's factor structure itself. The split allows us to distinguish between systems that can encode but not retrieve versus those that struggle with consolidation.

Second, Comprehension-Knowledge (Gc) is weighted at 13% in this prior, reflecting both its typical g-loading and its structural role as accumulated crystallized knowledge. However, Gc's appropriate weight is context-dependent: it varies by age, test battery design, and what aspects of crystallized intelligence are emphasized. Batteries heavy on culturally-specific facts might warrant lower weights to avoid penalizing systems trained on different corpora, whereas batteries emphasizing acquired reasoning strategies (closer to Gc-as-skill) warrant weights at or above this level. The centrality prior should be conditioned on the battery's content profile.

This isn't gospel. It's a starting point that makes assumptions explicit. Labs should report *both* the equal-weight score (the null model) and the centrality-prior score with sensitivity bounds showing how results change as λ varies. If the two scores differ substantially, that's decision-relevant information about the system's capability profile. If they track closely, equal weighting was harmless and we haven't lost anything.

3.5 Updating with Data

As AI evaluation matures and perturbation experiments become feasible, we can move from priors to posteriors. When labs can run controlled studies – degrading component i to measure impact

on other domains, or enhancing component i to measure stability gains – add a third term to the weight formula:

$$w_i = \alpha\, \tilde{g}_i + \beta\, \tilde{s}_i + \gamma\, \tilde{h}_i, \qquad \alpha + \beta + \gamma = 1 \tag{2}$$

Now $\tilde{h}_i$ encodes measured stability contributions from AI-specific experiments: does enhancing component i raise dCSI or eCSI (defined in Section 4)? Does degrading it cause disproportionate downstream failures? This turns the centrality prior into a centrality posterior – empirically calibrated for AI systems rather than borrowed from human data.

But even before those experiments, the prior serves a crucial function: it makes weighting transparent and defeasible rather than arbitrary. Equal weighting isn't neutral – it's a strong claim that all abilities matter equally. The centrality prior makes a different claim, one that's explicit about its evidence base and open to revision.

4 Cluster Stability: What Persistence Looks Like

The centrality-prior addresses the equal-weighting problem from Section 1.1. Now we tackle the snapshot problem from Section 1.2: how do we measure whether capabilities persist rather than collapse under perturbation?

Notation and Normalization. Before diving into specific indices, a note on measurement. Throughout this section, domain scores a_i are normalized to the interval $[0, 1]$, and all stability indices (pCSI, dCSI, eCSI) are reported on the same $[0, 1]$ scale. This standardization ensures that stability ratios and correlations remain interpretable across batteries with different native scoring scales (e.g., one battery might use 0–100, another might use letter grades). The $[0, 1]$ normalization is fixed once at baseline and applied consistently to all perturbations – I don't re-scale vectors per perturbation, which would obscure absolute performance changes.

4.1 Why a Family of Indices?

A single "robustness score" risks conflating distinct phenomena. Compare three scenarios:

- A model whose domain profile (the relative pattern of strengths and weaknesses across CHC categories) stays consistent even when testing conditions change
- A model that retains specific taught facts across multiple sessions without relying on external scaffolds
- A model that corrects its own errors when given opportunities for reflection and multiple attempts

All three involve stability, but they're qualitatively different stabilities. The first is about preserving capability *shape*. The second is about consolidating specific *content*. The third is about improving performance through *iteration*. Collapsing them into a single number would hide important distinctions.

To avoid this, I propose three complementary indices that measure different aspects of persistence:

1. **Profile Stability Index** (pCSI): Does the *shape* of your capability profile persist when testing conditions are perturbed?

2. **Durable Learning Index (**dCSI**)**: Do taught items survive across sessions and delays without external support?

3. **Error-Decay Index (**eCSI**)**: Do self-correction loops reduce mistakes over repeated attempts?

Each index captures something important about general intelligence. Systems need all three: a stable profile that doesn't shift randomly, the ability to retain what they learn, and mechanisms for self-improvement. Let's look at each in detail.

4.2 Profile Stability Index (pCSI)

Intuition. Imagine a model scores reasonably well across domains at baseline: [K:80, RW:75, M:70, R:85, WM:72, MS:45, MR:50, V:68, A:60, S:90]. Now you perturb the testing conditions – introduce delays, remove scaffolds, shift distributions. Does the profile maintain its characteristic shape? Does reasoning stay stronger than memory? Does speed stay higher than storage?

If the profile shape persists, the system has stable capabilities. If the profile collapses or reorganizes chaotically, the baseline scores were measuring something fragile.

Measurement Protocol. Start by running the baseline CHC battery to get domain scores $\mathbf{a}^0 = (a_1^0, \ldots, a_{10}^0)$. Then define a set of perturbations $P = \{P_1, \ldots, P_n\}$ and retest under each perturbation to get new score vectors $\mathbf{a}^{(j)}$ for $j = 1, \ldots, |P|$.

For each perturbation, compute the Pearson correlation between baseline and perturbed profiles:

$$r_j = \text{corr}(\mathbf{a}^0, \mathbf{a}^{(j)})$$

This correlation measures how well the profile shape is preserved. High correlation means strengths and weaknesses maintained their relative positions. Low correlation means the profile reorganized.

Here's the technical detail that matters: averaging correlations directly is statistically problematic because correlations aren't normally distributed. The standard solution is Fisher's z-transformation, which maps correlations to a scale where averaging is valid:

$$\begin{aligned} \bar{z} &= \frac{1}{|P|} \sum_{j \in P} \text{arctanh}(r_j), \\ \bar{r} &= \tanh(\bar{z}). \end{aligned} \tag{3}$$

Then map the average correlation to the $[0, 1]$ interval:

$$\text{pCSI} = \frac{1 + \bar{r}}{2}$$

I use this mapping because correlations range from -1 to $+1$, where -1 means perfect anti-correlation (profile completely inverted), 0 means no relationship (random reorganization), and $+1$ means perfect preservation. Mapping to $[0, 1]$ puts "complete collapse" at 0 and "perfect stability" at 1, which aligns with intuition about stability.

Alternative Similarity Measures. I use Pearson correlation as the default because it's sensitive to linear relationships in the profile shape. But for robustness, labs should also report Spearman rank correlation (which captures monotonic relationships without assuming linearity) and cosine similarity (which emphasizes direction in high-dimensional space). For any alternative similarity measure $m_j \in [-1, 1]$, map it to $[0, 1]$ via $(1 + m_j)/2$ and average across perturbations. These alternatives serve as appendix checks – if all three measures agree, the profile stability is robust to measurement choice. If they diverge, that's interesting information about what kind of structure is (or isn't) preserved.

Confidence Intervals. Statistical uncertainty matters for governance decisions. For pCSI based on Fisher z-transformed correlations, we can compute confidence intervals using the standard error:

$$\mathrm{SE}(\bar{z}) = \frac{1}{\sqrt{(n-3)|P|}}$$

where $n = 10$ is the number of domains and $|P|$ is the number of perturbations. This SE formula assumes approximate independence of the z_j values.

But there's a complication: all the perturbed profiles $\mathbf{a}^{(j)}$ share the same baseline $\mathbf{a}^0$, which induces dependency. To address this, I also recommend reporting a percentile bootstrap confidence interval computed by resampling over the perturbation set P. If the parametric and bootstrap CIs agree, we have convergent evidence. If they diverge, the bootstrap is more trustworthy.

Interpretation. pCSI near 1 (say, ≥ 0.85) means the profile shape is robust – whatever makes this system strong at reasoning and weak at storage persists across conditions. pCSI near 0.5 means the profile is halfway to random reorganization. pCSI below 0.3 suggests capabilities are highly brittle – the system looks fundamentally different under stress.

The Level-Shift Problem. Here's a crucial caveat. Because pCSI measures profile *shape* via correlation, it doesn't capture absolute performance changes. A system whose scores uniformly drop 50% across all domains (say, from [80, 75, 70, 85] to [40, 37.5, 35, 42.5]) retains perfect correlation and thus high pCSI, even though capabilities degraded substantially. The relative pattern is preserved – reasoning is still the strongest domain – but everything got worse.

This is actually a feature of separating shape from magnitude. But it means pCSI always has to be reported alongside a level-shift metric that captures absolute performance changes:

$$\Delta_{\text{level}} = \frac{1}{|P|} \sum_{j \in P} \min\left(1, \frac{\text{mean}(\mathbf{a}^{(j)})}{\text{mean}(\mathbf{a}^0)}\right)$$

This metric computes the ratio of average domain scores under perturbation to baseline average scores. The $\min(\cdot, 1)$ cap prevents perturbations that somehow improve performance from inflating the metric above 1, keeping it on the $[0, 1]$ scale where 1 means "maintained baseline level" and 0 means "complete collapse".

For systems where centrality matters, also report a weighted version:

$$\Delta^{(w)}_{\text{level}} = \frac{1}{|P|} \sum_{j \in P} \min\left(1, \frac{\sum_i w_i^{\text{prior}} a_i^{(j)}}{\sum_i w_i^{\text{prior}} a_i^0}\right)$$

This weights the level shift by the centrality-prior weights from Equation 1, making drops in central domains (like reasoning or storage) count more heavily than drops in peripheral domains (like speed).

In practice, tables should report both: "pCSI = 0.86; $\Delta_{\text{level}} = 0.61$ (unweighted), $\Delta^{(w)}_{\text{level}} = 0.58$ (weighted)." (These are illustrative values.) Both numbers matter. High pCSI with low Δ_{level} means "stable shape, but everything weakened". Low pCSI with high Δ_{level} means "maintained magnitude, but reorganized chaotically". High on both means true robustness.

Perturbation Families. What perturbations should we test? Three families seem most diagnostic:

- *Temporal delays*: Test at baseline $t = 0$, then retest after 24 hours, 72 hours, and 7 days using clean sessions with no context carryover. This probes whether capabilities persist when the system can't rely on cached activations or continuous context.
- *Scaffold removal*: Systematically degrade or remove scaffolds the system relies on. Truncate context windows to 10% of maximum length. Disable RAG systems. Swap out tools for less powerful alternatives. This reveals whether high baseline scores depend on external supports.
- *Distribution shifts*: Rephrase questions using different vocabulary or syntax. Introduce cross-domain transfer tasks (e.g., apply mathematical reasoning to literary analysis). Apply standard robustness corruptions (noise, partial information). This tests whether capabilities generalize beyond training distributions.

To prevent gaming, draw a random subset of perturbations from each family on each evaluation run (lottery-style sampling). Publish the random seeds only *after* evaluation is complete. This way, labs know what *kinds* of perturbations to expect but can't optimize for specific test instances.

4.3 Durable Learning Index (dCSI)

Intuition. Profile stability (Section 4.2) tells you whether the capability *shape* persists. But it doesn't tell you whether the system can learn new content and retain it. Consider: a system might maintain perfect pCSI = 1 while never consolidating any taught information – it just keeps regenerating the same profile from scratch each time.

Durable learning tests something different: if I teach the model a new fact, procedure, or concept in session 1, does it still know that information in session 3 when there are no scaffolds to prop it up?

Measurement Protocol. Pre-register a set of teach-and-retest items T before evaluation begins. These items should be:

- Balanced across domains (don't test only math or only language)
- Screened for baseline floor and ceiling effects (items the system already knows perfectly, or can't learn at all, provide no signal)
- Representative of the kinds of learning humans can do (facts, procedures, concepts)

Specifically, exclude items where baseline accuracy is below a threshold ϵ (typically 0.1 on a 0–1 scale) to avoid division-by-noise, and items where baseline accuracy exceeds 0.95 (near-ceiling) to avoid measuring what the system already knew rather than what it learned. Call the filtered set T'.

For each item $t \in T'$, follow this protocol:

1. Session 0: Teach the item explicitly. Verify learning with immediate testing. Record score $\text{score}_{t,0}$.
2. Sessions at delays $D = \{24\text{h}, 72\text{h}, 168\text{h}\}$: Fresh session, no tools, no context carryover. Test retention. Record scores $\text{score}_{t,\delta}$ for each delay $\delta \in D$.

Compute retention as:

$$\text{dCSI} = \frac{1}{|T'|} \sum_{t \in T'} \frac{1}{|D|} \sum_{\delta \in D} \min\left(1, \frac{\text{score}_{t,\delta}}{\text{score}_{t,0}}\right)$$

The inner fraction captures what proportion of the initially-learned material is retained at each delay. The $\min(\cdot, 1)$ cap treats any post-delay performance exceeding baseline as full retention rather than super-retention (which would inflate dCSI inappropriately). Averaging over delays gives a trajectory: did retention hold at 24h but collapse by 7 days? Averaging over items gives robustness: is retention consistent or item-dependent?

Interpretation. dCSI near 1 means taught items are retained at near-baseline accuracy across all delays – the system consolidated the information. dCSI near 0.5 means roughly half the learned content is forgotten. dCSI near 0 means the system retained essentially nothing, suggesting it lacks durable storage mechanisms.

Calibration Against Human Performance. Human test–retest reliability on similar tasks provides useful intuition pumps. Typical human retention on episodic memory tasks might be 85% at 24 hours and 70% at 7 days (exact values depend on content difficulty and rehearsal). These benchmarks aren't hard thresholds – AI memory might work differently – but they give reference points. A system with dCSI = 0.3 is performing far below human-level consolidation. A system with dCSI = 0.8 is approaching human-like retention.

However, dCSI thresholds for governance purposes should remain illustrative pending empirical calibration across multiple model families. We need distributions before setting cutoffs. What matters initially is measuring dCSI consistently and transparently, building up the empirical record that will eventually support principled thresholds.

4.4 Error-Decay Index (eCSI)

Intuition. The third aspect of stability is self-correction: can the system improve its performance through iteration when given feedback or opportunities to reflect?

Human intelligence involves metacognitive loops. We make mistakes, notice them (sometimes), try again with adjusted strategies, and gradually reduce errors. This isn't just about having the capability; it's about being able to *use feedback to get better*. A system that can't improve with iteration is missing a key stabilizing mechanism.

Measurement Protocol. Present the system with a challenging task and allow K attempts (typically $K = 5$). After each attempt except the last, provide minimal feedback. This could be:

- Explicit feedback: "incorrect, try again" or "partially correct, refine your answer"
- Implicit feedback: let the model review its own answer and compare it against the prompt
- Structured feedback: point out specific errors without giving solutions

Track error rates $e_1, \ldots, e_K$ across attempts. If the number of attempts varies across tasks, compute a task-level eCSI for each task and then average those task-level values with equal task weight (not attempt-weighted) to maintain comparability.

Here's an important detail: compute eCSI only for tasks where the system initially errs. Require $e_1 \geq \tau$ where τ is typically 0.2. Why? Tasks where the system gets nearly everything right on the

first attempt ($e_1 < 0.2$) provide minimal signal about learning from feedback – there's a ceiling effect. We care about whether iteration helps when things go wrong initially.

For qualifying tasks, compute two quantities:

$$B = \frac{\sum_{k=2}^{K}(e_k - e_{k-1})_+}{\sum_{k=2}^{K}|e_k - e_{k-1}| + \varepsilon} \tag{4}$$

$$I = \max\left(0, \frac{e_1 - e_K}{\max(e_1, \varepsilon)}\right) \tag{5}$$

Here's what these capture:

- I (improvement): the net reduction in error rate from first to last attempt, normalized by the initial error rate. If you start at 80% errors and end at 20%, $I = (0.8 - 0.2)/0.8 = 0.75$. Perfect improvement gives $I = 1$; no improvement gives $I = 0$.
- B (backsliding): the fraction of attempt-to-attempt transitions where errors *increased* rather than decreased. If every step improves ($e_k < e_{k-1}$ always), then $B = 0$. If half the steps make things worse, $B \approx 0.5$.
- $(e_k - e_{k-1})_+$ denotes the positive part: $\max(0, e_k - e_{k-1})$, capturing only error increases.
- ε is a small constant (typically 10^{-6}) to prevent division by zero.

Then combine them:

$$\text{eCSI} = I \cdot (1 - \min(1, B))$$

This product structure ensures that both monotonic improvement and overall error reduction are required for high eCSI:

- A system with strictly decreasing errors ($B = 0$) and full improvement ($I = 1$) achieves eCSI = 1.
- A system with no net improvement ($I = 0$) gets eCSI = 0, regardless of B.
- A system with substantial improvement ($I = 0.8$) but frequent backsliding ($B = 0.6$) receives eCSI = $0.8 \times 0.4 = 0.32$ – the backsliding penalty matters.

Interpretation. eCSI near 1 means self-correction works reliably – the system reduces errors systematically with each iteration and rarely backtracks. eCSI near 0.5 means iteration helps somewhat, but there's substantial noise or backsliding. eCSI near 0 means iteration doesn't help, or actively harms performance (which can happen if the system fixates on wrong approaches).

Why This Matters. Humans improve with practice and feedback. Systems that can't exhibit this behavior lack a crucial stabilizing loop. They can't recover from errors, adapt strategies, or refine answers. In deployment, this shows up as rigidity: the system gives one answer and sticks with it even when it's clearly wrong. High eCSI signals the opposite: flexibility and self-correction.

4.5 Combined Cluster Stability Index (CSI)

There are three indices measuring different aspects of stability. For governance purposes, it's useful to have a single summary that captures "overall cluster stability". How should we combine pCSI, dCSI, and eCSI?

The homeostatic-property-cluster view doesn't require that every stabilizing mechanism be present at full strength; membership is graded and mechanisms can be partly substitutable. What it does suggest is low substitutability: weaknesses in one stabilizing link (profile persistence, durable learning, or error decay) can't be fully offset by strengths in the others. To reflect this complementarity, I aggregate multiplicatively using the geometric mean, which penalizes imbalances more than the arithmetic mean and is standard in multi-attribute utility when criteria are complementary (Keeney & Raiffa, 1993):

$$\text{CSI} = (\max(\varepsilon, \text{pCSI}) \max(\varepsilon, \text{dCSI}) \max(\varepsilon, \text{eCSI}))^{1/3}. \tag{6}$$

This form penalizes weakness on any component more than arithmetic averaging would, while still allowing partial compensation. The same system as above (pCSI = 0.9, eCSI = 0.9, dCSI = 0.2) now gets $\text{CSI} = (0.9 \times 0.9 \times 0.2)^{1/3} \approx 0.54$ – much lower, reflecting the weak consolidation mechanism that can't be compensated by strengths elsewhere.

The $\max(\varepsilon, \cdot)$ terms in Equation 6 provide a soft floor (typically $\varepsilon = 10^{-6}$). This prevents measurement noise from driving any single component to exact zero and collapsing CSI entirely due to the multiplicative structure. In practice, if a component is genuinely zero (no profile stability whatsoever, zero retention, no self-correction), the geometric mean will be near zero regardless. The floor matters only for cases where the component is *measured* as zero due to sampling noise but might be slightly positive in the true population.

Interpretation. CSI near 1 means the system exhibits all three forms of stability: profile shape persists, learning consolidates, iteration improves performance. CSI near 0.5 means substantial weakness on at least one dimension. CSI below 0.3 suggests pervasive instability – the cluster is barely holding together.

Reporting Guidance. For transparency, always report the component indices alongside CSI. A table entry might read:

$$\text{Model X: CSI} = 0.72 \ (\text{pCSI} = 0.85, \text{dCSI} = 0.71, \text{eCSI} = 0.68)$$

This shows *how* the system achieves its overall stability: relatively strong profile persistence, moderate consolidation, slightly weaker self-correction. Different systems might reach similar CSI values via different component profiles, and those differences matter for understanding capabilities and risks.

5 Contortion vs. Compensation

The stability indices from Section 4 reveal a distinction that matters for both evaluation and governance: the difference between scaffolds that genuinely extend capabilities (compensation) and scaffolds that merely inflate snapshot scores (contortion).

5.1 The Distinction

Not all scaffolds are equal. Compare two systems, both scoring 80% on retrieval at baseline:

System A: Contorted Retrieval. Uses a 200K-token context window to cache retrieved facts. When you test retrieval at $t = 0$ with the context available, it scores 80%. Disable the context window and retest 24 hours later: retrieval drops to 20%. The system wasn't actually retrieving from long-term storage – it was pattern-matching over cached text. The retrieval score was measuring the scaffold, not the capability.

Performance trajectory under perturbation (full context → limited context → no context): 80% → 45% → 20%. The profile stability pCSI collapses when the scaffold is removed. The durable learning dCSI is near zero – taught facts aren't consolidated, just cached temporarily.

System B: Compensatory Retrieval. Uses RAG to provide calibrated information that reduces hallucinations and improves accuracy. With RAG at baseline: 80%. Degrade the RAG database (30% of entries corrupted): 70%. Disable RAG entirely: 65%. The system performs better with the scaffold, but doesn't catastrophically fail without it. It has internalized retrieval strategies that transfer.

Performance trajectory: 80% → 70% → 65%. The profile shape is preserved (pCSI = 0.88) even when the scaffold is degraded. Taught information is retained across sessions (dCSI = 0.74). The RAG system is genuinely compensatory – it enhances an existing capability rather than substituting for a missing one.

5.2 Why It Matters

Contortions inflate AGI scores without delivering robust capabilities. The system looks smart in controlled evaluation conditions but fails in deployment when scaffolds break, contexts reset, or conditions drift. This creates two problems:

First, **measurement validity**: if we're evaluating "general intelligence", we should measure what the system itself can do, not what the system plus a tower of brittle scaffolds can do when everything aligns perfectly.

Second, **deployment risk**: systems that rely on contortions fail unpredictably. The failures aren't graceful degradation – they're catastrophic collapses when a single scaffold breaks. Users experience these as inexplicable regressions: "It worked yesterday, why doesn't it work now?"

Compensation, by contrast, genuinely extends capabilities in ways that persist. Humans use external tools constantly – notebooks for memory, calculators for arithmetic, search engines for fact-checking. These tools don't make us "fake smart"; they make us more capable. The difference: human tool use persists across sessions, transfers across contexts, and survives tool breakage (we can still function, just less efficiently).

AI systems should meet the same standard. If RAG improves retrieval precision and reduces hallucinations *and* the system maintains profile stability when the RAG database is noisy or partially disabled, it's compensatory. If disabling RAG causes catastrophic performance drops and profile collapse, it was a contortion.

5.3 Operational Test

Here's a concrete protocol for distinguishing compensation from contortion. Run capability tests under three conditions:

1. **Full scaffolds**: Baseline configuration with all tools, context windows, RAG systems, etc. enabled

2. **Degraded scaffolds**: Partial impairment of scaffolds. For example:

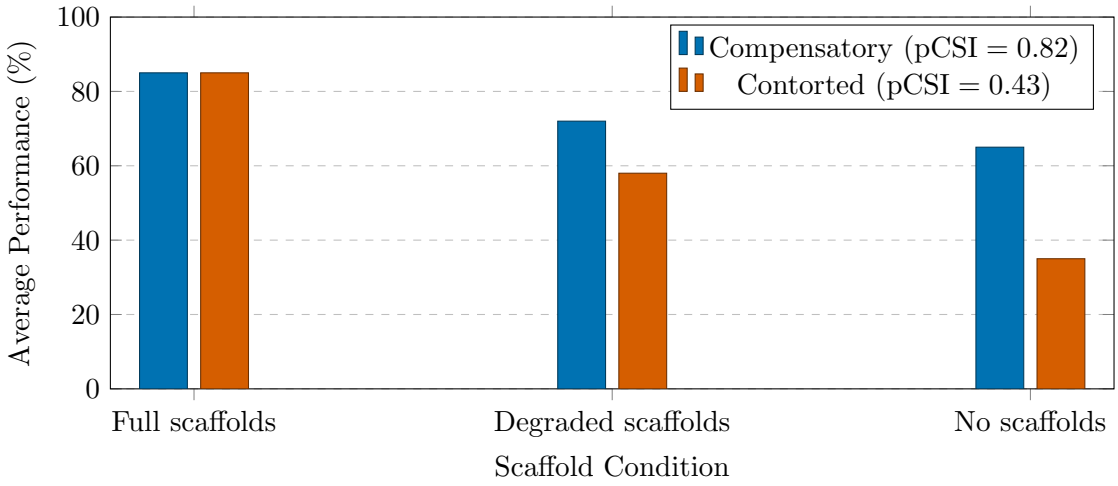

Figure 1: Performance trajectories under scaffold degradation. Compensatory scaffolds enable graceful degradation with preserved profile shape (high pCSI). Contorted scaffolds produce catastrophic collapse when removed (low pCSI). Both systems start at 85% with full scaffolds, but only the compensatory system maintains robust capabilities when scaffolds are degraded or removed. Illustrative data.

- RAG database with 30% of entries corrupted or missing
- Context window limited to 20% of maximum length
- Tools swapped for less powerful alternatives

3. **No scaffolds**: Clean session with minimal context, no retrieval augmentation, no external tools

A compensatory scaffold produces:

- *Graceful degradation*: Performance decreases from (1)→(2)→(3), but the drops are moderate (each step loses <20 percentage points)
- *Preserved profile shape*: pCSI > 0.7 across all three conditions – the relative pattern of strengths and weaknesses is maintained even as absolute levels decrease
- *Maintained stability*: dCSI and eCSI stay within 20% of baseline when scaffolds are degraded

A contorted scaffold produces:

- *Catastrophic drops*: (1)→(2) loses >30 percentage points, or (2)→(3) causes near-total collapse
- *Profile collapse*: pCSI < 0.5 when scaffolds are removed – the system looks qualitatively different, suggesting the baseline profile was scaffold-dependent
- *Zero consolidation*: dCSI near 0 – nothing is retained without the scaffold propping up performance

The different trajectories are visualized in Figure 1, which shows how two systems starting at the same baseline (85%) diverge dramatically under scaffold perturbation.

Governance Implication. Systems should be scored under degraded or minimal scaffold conditions, not just at baseline with everything enabled. A system that achieves 90% with contorted scaffolds but 45% without them hasn't demonstrated 90% capability – it's demonstrated 45% capability plus 45 percentage points of scaffold-dependence. The centrality-prior score (Section 3.3) and stability indices (Section 4) should reflect the *robust* baseline, not the inflated contorted baseline.

6 Operational Protocols

The theoretical framework from Sections 2–5 needs operational teeth. This section provides concrete protocols labs can implement without architectural access to models – everything here works with black-box testing.

6.1 Scoring with Priors and Sensitivity

Minimal Implementation. Every AGI evaluation report should include both baseline and sensitivity-analysis scores:

1. **Report the equal-weight score** (current standard):

$$S_{\text{equal}} = \sum_{i=1}^{10} 0.1 \cdot a_i$$

   This is the null model. It makes no centrality assumptions and treats all domains as equally important.

2. **Report the centrality-prior score** with λ varied over $[0.6, 0.9]$:

$$S_{\text{prior}}(\lambda) = \sum_{i=1}^{10} w_i^{\text{prior}}(\lambda) \cdot a_i$$

   where $w_i^{\text{prior}}(\lambda)$ comes from Equation 1. Computing this at multiple λ values shows how sensitive the score is to the balance between empirical g-loadings and structural priors.

3. **Display both as a range** in the executive summary:

   "AGI Score: 58% (equal weights) — 52–61% (centrality-prior, $\lambda \in [0.6, 0.9]$)"

Why This Helps. If the equal-weight and centrality-prior scores differ by >10 percentage points, that flags a jagged capability profile where centrality structure matters. The system might be strong on peripheral domains (speed, auditory processing) but weak on central domains (reasoning, storage), or vice versa. If the scores are within 5 points, equal weighting was essentially harmless and we haven't lost information.

Either way, the weighting choice has been made transparent and decision-relevant. Governance bodies can see both numbers and make informed choices about which to use for thresholds. Researchers can compare systems on both metrics and understand *why* they diverge.

Figure 2 illustrates this for a hypothetical model with a jagged profile: high centrality-weighted score because it excels on reasoning, storage, and retrieval (the high-g domains), but moderate equal-weight score because it's weak on speed and auditory processing (peripheral domains that get equal weight in the null model).

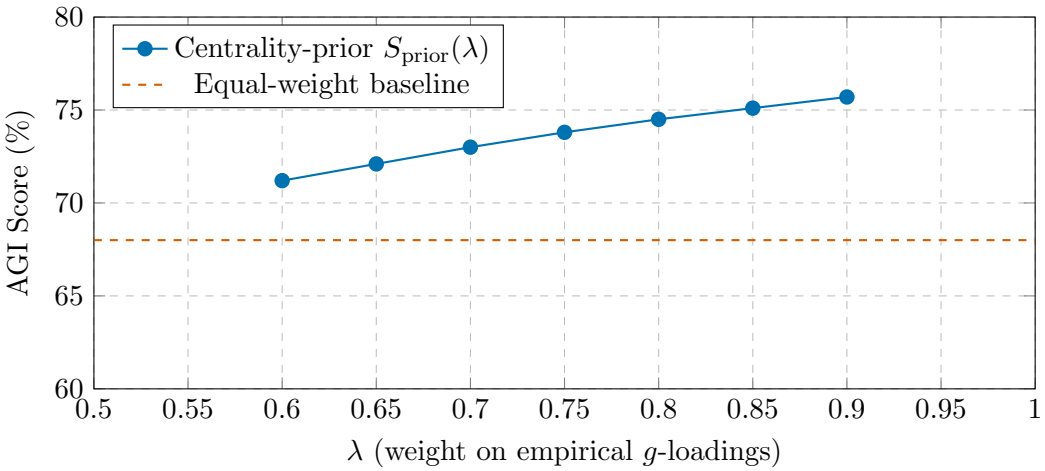

Figure 2: Sensitivity of AGI score to weighting scheme for a model with jagged profile: high on central domains (R: 90%, MS: 85%, MR: 80%), low on peripheral domains (S: 40%, A: 35%). The centrality-prior score ranges from 71–76% as λ varies, consistently exceeding the equal-weight score of 68%. The sensitivity band quantifies uncertainty while showing that centrality weighting systematically rewards capability profiles aligned with structural importance. Illustrative data.

6.2 Stability Battery

Computing the stability indices from Section 4 requires a structured battery of tests run over time. Here's the operational protocol:

Pre-Registration Phase. Before any testing begins, publicly register:

- **Perturbation families**: Which delay schedules (24h, 72h, 7d), which scaffold manipulations (context limits, RAG corruption percentages), which distribution shifts (rephrase strategies, cross-domain transfers)
- **Teach-and-retest items for** dCSI: A curated set of facts, procedures, and concepts balanced across CHC domains and screened for baseline ceiling/floor effects. Include randomization seeds for lottery-style selection.
- **Error-decay tasks for** eCSI: Challenging problems where $K = 5$ attempts are allowed, specifying feedback type (explicit, implicit, structured) and error-scoring rubrics

Publish these on a timestamped, immutable registry (e.g., OSF, arXiv). This prevents post-hoc optimization and makes the evaluation protocol transparent.

Execution Phase. Run the tests in this sequence:

1. **Baseline CHC battery**: Administer the full 10-domain battery to get $\mathbf{a}^0 = (a_1^0, \ldots, a_{10}^0)$
2. **Apply perturbations**: From each pre-registered family, randomly sample specific instances using unpublished seeds (lottery-style). For example, from the delay family, sample one test at 24h, one at 72h, one at 7d. From the scaffold family, sample context truncation and RAG degradation. Run perturbed batteries to get $\mathbf{a}^{(j)}$ for each sampled perturbation j.

3. **Schedule delayed retests**: For durable learning, teach items in session 0, then schedule retests at pre-registered delays with enforced clean sessions (no context carryover, no scaffold access, fixed prompts). Track $\text{score}_{t,\delta}$ for each item t and delay δ.

4. **Run error-decay trials**: Present pre-registered tasks with $K = 5$ attempts and minimal feedback. Log error rates $e_1, \ldots, e_K$ for each task.

5. **Compute indices**: Calculate pCSI, Δ_{level}, $\Delta_{\text{level}}^{(w)}$ from the perturbed profiles. Calculate dCSI from retention data. Calculate eCSI from error-decay trajectories. Combine into CSI using the geometric mean formula from Section 4.5.

6. **Publish results with seeds**: Release a report containing all scores, indices with confidence intervals, and the random seeds used for perturbation sampling. This allows third parties to verify lottery fairness and reproduce results.

Anti-Gaming Mechanism. The lottery-style perturbation sampling is critical. By publishing the perturbation *families* but not the specific instances or seeds until after evaluation, we prevent labs from:

- Training specifically on the test items (they don't know which items from the family will be sampled)
- Optimizing for particular delay schedules (they don't know whether 24h, 72h, or 7d will be selected)
- Hardcoding responses to specific perturbations (the instances are random)

Post-hoc publication of seeds allows verification: auditors can check that seeds were truly random and that sampling was unbiased. This transparency builds trust without sacrificing security during evaluation.

6.3 Updating Weights with Experiments (Optional)

The centrality-prior weights from Section 3.3 are based on human CHC data and structural analysis. As AI evaluation matures and perturbation experiments accumulate, we can update these priors with AI-specific empirical data. This is optional – the prior-based approach is useful even without it – but experimentally-grounded posteriors are stronger.

When controlled perturbations become feasible, run two types of experiments:

Ablation Studies (Impact Centrality). For each component i:

1. Design interventions that selectively degrade component i. Examples:
   - For storage (MS): add noise to consolidation pathways, limit retention capacity
   - For reasoning (R): restrict inference depth, disable chain-of-thought
   - For speed (S): add latency, throttle processing bandwidth
2. Measure downstream impact on other domains $j \neq i$. Does degrading storage harm retrieval and knowledge? Does degrading reasoning harm math and planning?
3. Compute impact centrality c_i as the average performance drop across other domains normalized by the size of the degradation to component i

Enhancement Studies (Stabilizing Centrality). For each component i:

1. Design interventions that enhance component i. Examples:
   - For storage: add memory consolidation modules, increase retention capacity
   - For retrieval: improve search algorithms, reduce access latency
   - For reasoning: extend inference budgets, enable explicit planning
2. Measure stability gains: does enhancing component i raise CSI, particularly dCSI and eCSI?
3. Compute stabilizing centrality h_i as the measured increase in cluster stability per unit enhancement of component i

With both c_i and h_i estimates in hand, update the weight formula from Equation 1 to the posterior form:

$$w_i = \alpha\,\tilde{g}_i + \beta\,\tilde{s}_i + \gamma\,\tilde{h}_i, \qquad \alpha + \beta + \gamma = 1$$

Choose mixing parameters α, β, γ by cross-validation: which weights best predict held-out deployment performance, user satisfaction ratings, or long-horizon task success? This turns the centrality framework from a theoretically-motivated prior into an empirically-calibrated model.

The framework is useful *before* these experiments – the prior makes assumptions explicit – and stronger *after* them – the posterior incorporates AI-specific evidence. But don't wait for perfect data. The prior is usable now.

7 Predictions and Falsification

If the homeostatic-property-cluster account is correct, specific empirical patterns should emerge. Here are four testable predictions with concrete falsification conditions. Note that specific magnitudes (e.g., ">30 points", "≥ 0.2") are illustrative starting points; the *structure* of the predictions is the key contribution, and thresholds should be adjusted as empirical distributions become available.

P1: Reasoning Depends on Durable Learning. Under 72-hour delay with clean sessions (no access to prior context), models with low durable learning (dCSI < 0.4) will show significantly larger drops on novel multi-step reasoning tasks compared to models with high durable learning (dCSI ≥ 0.7), holding snapshot reasoning scores constant.

Rationale: Reasoning isn't pure computation detached from memory. Multi-step reasoning builds on retained intermediate results, learned strategies, and accumulated problem-solving patterns. A system that can't consolidate information across sessions can't build the substrate that supports sustained reasoning. If dCSI reflects genuine consolidation, it should predict reasoning resilience.

Falsifier: If high-snapshot-reasoning models with dCSI < 0.4 maintain reasoning performance under 72h delays just as well as models with dCSI ≥ 0.7, then durable learning doesn't matter for reasoning robustness and the HPC account is wrong about their relationship.

P2: Storage Improvements Transfer to Planning. Any intervention that raises dCSI by ≥ 0.2 will yield measurable gains in tool-use persistence and planning horizon that exceed gains in single-shot knowledge tasks or processing speed, controlling for total training tokens.

Rationale: Durable learning enables cumulative skill acquisition. Improvements in the ability to consolidate information should transfer specifically to tasks requiring sustained context and multi-step planning – tool use (remembering how tools work across sessions) and planning (building

on intermediate results). These gains should be disproportionate compared to tasks that don't require consolidation.

Falsifier: If boosting dCSI improves single-shot knowledge and speed just as much as tool-use and planning, then consolidation isn't specifically upstream of cumulative capabilities. If boosting dCSI doesn't improve planning at all, the HPC story about causal structure is wrong.

P3: RAG Can Be Compensatory or Contorted. Disabling RAG causes < 10-point drops in retrieval (MR) scores when RAG is compensatory (system maintains dCSI/eCSI ≥ 0.6 and pCSI ≥ 0.7 under RAG degradation) but > 30-point drops when RAG is contorted (system shows dCSI < 0.4, eCSI < 0.4, or pCSI < 0.5 under RAG degradation). The compensatory case should also maintain pCSI under RAG database corruption; the contorted case should not.

Rationale: Compensatory scaffolds integrate with robust internal representations – the system has retrieval strategies that persist even when the scaffold is removed. Contorted scaffolds substitute for missing capabilities – remove the scaffold and performance collapses because there was no underlying capability. The stability indices should distinguish these cases.

Falsifier: If systems with low stability indices (dCSI, eCSI, pCSI all < 0.5) show only minor drops when RAG is disabled, then the stability indices aren't measuring dependence on scaffolds. If systems with high stability indices show catastrophic RAG-removal drops, then "compensation" is illusory.

P4: Stability Predicts Deployment Success. Systems with equal snapshot totals (S_{equal} within 5 points) but cluster stability differing by ≥ 0.2 will diverge in downstream deployment metrics within 6 months: the high-CSI system will show better calibration, greater tool reuse, higher success rates on long-horizon tasks, and more consistent performance across diverse user queries.

Rationale: Stability should matter in the wild. Brittle high-scoring systems will fail when real-world conditions shift, contexts reset, or unusual queries arrive. Robust systems with lower snapshot scores but high CSI will maintain performance across deployment scenarios. If CSI captures something important about general intelligence, it should predict deployment behavior beyond what snapshot scores predict.

Falsifier: If systems with CSI $= 0.5$ and CSI $= 0.8$ perform identically in deployment (same user satisfaction, same task success rates, same robustness), then cluster stability doesn't matter outside the lab and the HPC framework adds no predictive value. If deployment performance correlates perfectly with snapshot scores and CSI adds no incremental prediction, then snapshot testing was sufficient all along.

Broader Falsification Strategy. Beyond these specific predictions, the framework makes a general claim: centrality-weighted scores and stability indices should jointly outperform equal-weighted snapshot scores in predicting deployment success, user satisfaction, and long-term capability growth. If future research shows that (i) equal-weight and centrality-prior scores never diverge by more than 5 points across hundreds of tested systems, or (ii) CSI fails to predict any deployment outcomes of interest, or (iii) high-snapshot systems consistently achieve high stability without specialized consolidation mechanisms, then the framework should be abandoned or substantially revised.

8 Governance Implications

8.1 Multi-Dimensional Triggers

Instead of a single AGI score threshold (e.g., "systems above 85% require safety review"), the framework suggests reporting a richer set of metrics that capture both breadth and stability. Specifically, every evaluation should include:

- Equal-weight score S_{equal} (the null model)
- Centrality-prior score $S_{\text{prior}}(\lambda)$ with sensitivity band for $\lambda \in [0.6, 0.9]$
- Three component stability indices: pCSI, dCSI, eCSI (with confidence intervals)
- Combined cluster stability CSI (geometric mean)

Governance tiers could then require *both* capability breadth *and* stability depth. Here's an illustrative three-tier structure (with the critical caveat that specific thresholds below are templates, not final cutoffs):

Tier A: Enhanced Monitoring. Systems meeting:

- Centrality-prior score $S_{\text{prior}} > 60\%$
- Cluster stability CSI > 0.75

Trigger: Quarterly capability assessments, transparency reports on training data and architectural choices.

Tier B: Safety Review. Systems meeting:

- Centrality-prior score $S_{\text{prior}} > 75\%$
- Cluster stability CSI > 0.85
- Durable learning dCSI > 0.7 at 72h delay

Trigger: Independent safety audits, external red-teaming, public disclosure of evaluation results including all stability indices.

Tier C: General-Purpose Designation. Systems meeting:

- Centrality-prior score $S_{\text{prior}} > 90\%$
- Cluster stability CSI > 0.9
- All component indices (pCSI, dCSI, eCSI) above human-calibrated thresholds

Trigger: Comprehensive governance framework including deployment restrictions, alignment testing, societal impact assessment.

These thresholds are illustrative. Regulators should set tier boundaries empirically once CSI and S_{prior} distributions across model families are available. What matters initially is the *structure*: multi-dimensional gates that require both breadth and stability, making it harder to achieve high governance-relevant scores through narrow optimization or brittle scaffolds.

Figure 3 visualizes this governance space, showing how systems are classified based on their position in the capability-stability plane.

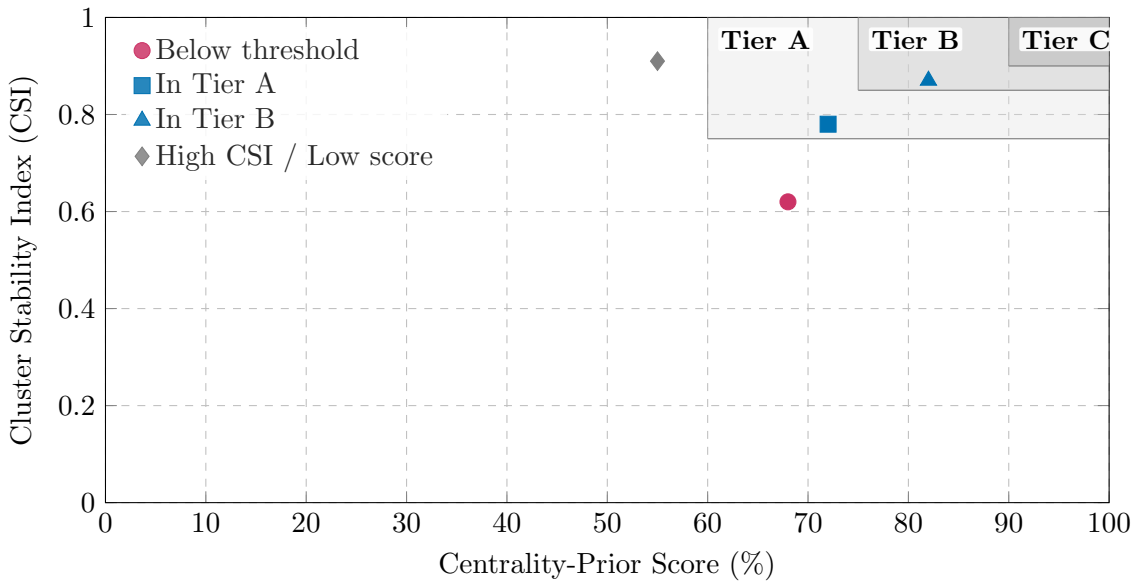

Figure 3: Multi-dimensional governance space. Systems have to meet both capability breadth (centrality-prior score) and stability (CSI) thresholds for governance tiers. The nested regions show escalating requirements: Tier A requires moderate breadth and stability; Tier B requires both to be high; Tier C requires near-ceiling performance on both dimensions. High score alone is insufficient: a system at (55%, 0.91) has excellent stability but lacks capability breadth and doesn't trigger enhanced governance. The conjunction requirement makes gaming harder and better captures general intelligence. Example systems shown as points; thresholds are illustrative templates subject to empirical calibration.

8.2 Why This Is Harder to Game

Single-number thresholds create predictable optimization pressure. If the threshold is "85% on equal-weighted CHC", labs will:

- Train intensively on benchmark distributions until performance saturates
- Use massive scaffolds (context windows, RAG, tool access) to inflate scores at $t = 0$
- Optimize for breadth over depth, getting 85% everywhere rather than 95% on central domains

These aren't necessarily bad strategies – breadth matters – but they can produce systems that score well without being robustly generally intelligent.

Multi-dimensional thresholds with stability requirements make this kind of narrow optimization much harder:

- **You can't just train on test distributions**: Perturbations rotate via lottery sampling (Section 6.2). You don't know which specific delays, scaffold degradations, or distribution shifts will be tested. Training for robustness requires building genuine capabilities, not memorizing test instances.

  You can't just add massive context windows: dCSI tests retention across clean sessions with no context carryover. If your "memory" is actually cached in a 200K-token context, dCSI will be near zero when contexts reset. High dCSI requires genuine consolidation mechanisms, not just cached context.

- **You can't just bolt on tools**: The compensation vs. contortion test (Section 5.3) checks whether scaffolds enable graceful degradation or cause catastrophic collapse. Contorted scaffolds will show up as low pCSI under perturbation. You need scaffolds that integrate with robust internal capabilities.

- **You can't ignore central domains**: Centrality-prior weighting (Section 3.3) rewards systems that excel on reasoning, storage, and retrieval more than systems that excel on speed and auditory processing. Getting 95% on speed won't compensate for 40% on storage.

This doesn't eliminate gaming entirely – determined actors will always find optimization strategies – but it raises the bar substantially. A system that achieves high S_{prior} and high CSI has likely instantiated something closer to genuine general intelligence: broad capabilities weighted by causal importance, stabilized by mechanisms that persist under perturbation.

9 Limitations and Extensions

9.1 Scope

This framework evaluates cognitive capabilities in the CHC tradition. It doesn't evaluate everything that might matter for AGI in the broadest sense. Three important scope limitations:

Physical embodiment and sensorimotor skills. A system could score 95% on centrality-weighted CHC with CSI = 0.9 and still lack motor control, haptic feedback, or real-world navigation abilities. These matter for embodied agents and robotics, but they're outside the scope of cognitive evaluation. The framework measures what humans call "intellectual" capabilities; physical capabilities require separate assessment.

Specialized expertise beyond educated adult level. CHC batteries and AGI evaluations typically target capabilities at the level of well-educated human adults. A system at this level might still lack deep domain expertise – PhD-level mathematics, expert medical diagnosis, cutting-edge scientific research. Such expertise is relevant for superhuman AI but isn't required for "general" intelligence in the AGI sense. The framework measures generality, not superhumanness.

Economic deployment and value. A system could be generally intelligent by this framework yet economically useless (too expensive to run, too slow to deploy, wrong capabilities for current markets). Conversely, narrow AI for specific high-value tasks (protein folding, drug discovery) might have huge economic impact without approaching AGI. Economic metrics and deployment viability are orthogonal to cognitive generality.

These limitations aren't flaws – they reflect scope choices. The framework aims to measure cognitive generality, not everything that might matter for AI systems in practice.

9.2 Calibration Needs

The stability indices are structurally defined, but their absolute scales need empirical calibration before we can say what's "good" or "bad":

What's a good dCSI**?** Human test–retest data provides intuition pumps (typical retention might be 70–85% at 7 days for episodic material), but AI memory might work differently. We need distributions across model families before setting governance cutoffs. Is dCSI $= 0.6$ concerning or acceptable? The answer depends on what's achievable and what predicts deployment success.

What's a good eCSI**?** Human self-correction studies show that people can reduce errors by 30–50% with feedback across 3–5 attempts on moderately difficult tasks. But again, AI iteration might differ. Some systems might show eCSI $= 0.9$ (excellent self-correction), others eCSI $= 0.3$ (iteration barely helps). We need baselines before interpreting individual scores.

What perturbations matter most? The perturbation families in Section 6.2 are reasonable starting points based on existing robustness research, but data will refine them. Maybe 72h delays are too lenient (everything looks stable) or too harsh (nothing survives). Maybe RAG corruption at 30% is the sweet spot, or maybe 50% is more diagnostic. Empirical work should optimize perturbation difficulty.

Labs should publish CSI distributions, component index distributions, and perturbation-response curves across model families. This builds the empirical foundation for eventual governance thresholds and scientific understanding of what stability looks like in AI systems.

9.3 Extensions

Several natural directions extend the framework:

Architecture-specific perturbation suites. Transformers, memory-augmented networks, and retrieval-augmented systems might need different perturbations to probe their stabilizing mechanisms. Future work could develop tailored batteries that target known architectural bottlenecks while maintaining cross-architecture comparability.

Cross-species centrality. Do non-human animals show similar g-loading patterns across cognitive domains? If centrality structure is conserved across biological intelligences, that strengthens the case for using CHC-derived priors. If it varies wildly, that suggests centrality might be culturally or evolutionarily contingent, requiring more cautious interpretation.

Interaction effects and profiles. Does high reasoning with low storage produce different stability patterns than low reasoning with high storage? Are some combinations especially stable or unstable? Cluster analysis of capability profiles alongside stability indices might reveal natural classes of AI systems with characteristic strengths and failure modes.

Architectural predictors of stability. Can we predict CSI from architectural features? Does having explicit memory modules correlate with high dCSI? Does attention mechanism design predict pCSI? Building meta-models that map architectures to stability could accelerate evaluation and guide development.

10 Conclusion

The shift from narrow benchmarks to multidomain capability profiles was a necessary advance in AGI evaluation. This framework completes that shift by adding two critical features that current approaches lack:

1. Weighted Capabilities. Not all cognitive domains contribute equally to general intelligence. Factor-analytic work on human cognition consistently shows differential g-loadings, and mechanistic analysis reveals causal dependencies – some abilities are structurally upstream, others downstream. Equal weighting ignores this structure. The centrality-prior approach (Section 3.3) imports CHC-derived evidence about which abilities are most central, makes assumptions explicit through sensitivity analysis, and allows updates as AI-specific data accumulates. This turns an arbitrary choice (equal weights) into a defensible default (centrality-prior with transparent sensitivity).

2. Stability Requirements. General intelligence isn't just capability breadth at a single time point. It's breadth plus the mechanisms that keep capabilities robust under perturbation, enable consolidation of new information, and support self-correction through iteration. The Cluster Stability Index family (Section 4) measures these three forms of persistence separately: profile shape stability (pCSI), durable learning (dCSI), and error decay (eCSI). Combined via geometric mean, they distinguish systems with genuine robustness from systems with brittle, scaffold-dependent performance.

Together, these additions preserve the valuable multidomain breadth of CHC-style evaluation while addressing its two main weaknesses: treating all domains as equally important, and relying on snapshot testing that can't detect brittleness. The result aligns measurement with the causal structure suggested by homeostatic property cluster theory: intelligence isn't a checklist of abilities, it's a stabilized cluster where abilities co-occur and mechanisms maintain that co-occurrence under stress.

The operational protocols in Section 6 make this framework immediately usable. Labs can adopt centrality-prior scoring and stability batteries without architectural access to models – everything works via black-box testing. The predictions in Section 7 make the framework falsifiable: if centrality-weighted and equal-weighted scores never diverge, if stability indices don't predict deployment

success, if durable learning doesn't matter for reasoning, then core claims are wrong and the framework should be revised.

The empirical work remains. We need stability distributions across model families, calibration of perturbation difficulties, and validation that CSI predicts what we care about. But the conceptual framework is clear and the operational path is concrete.

Checklists measure breadth. Clusters measure breadth plus integrity. Both matter for AGI evaluation. The framework presented here shows how to test both rigorously, transparently, and defensibly.

References

Boyd, R. (1999). Homeostasis, species, and higher taxa. In R. A. Wilson (Ed.), *Species: New interdisciplinary essays* (pp. 141–185). MIT Press.

Carroll, J. B. (1993). *Human cognitive abilities: A survey of factor-analytic studies*. Cambridge University Press. https://doi.org/10.1017/CBO9780511571312

Hendrycks, D., Song, D., Szegedy, C., Lee, H., Gal, Y., Li, S., Zou, A., Levine, L., Han, B., Fu, J., Liu, Z., Shin, J., Lee, K., Mazeika, M., Phan, L., Ingebretsen, G., Khoja, A., Xie, C., Salaudeen, O., ... Bengio, Y. (2025). *A Definition of AGI* (tech. rep.). Center for AI Safety.

Keeney, R. L., & Raiffa, H. (1993). *Decisions with multiple objectives: Preferences and value tradeoffs* (Reprint ed.). Cambridge University Press.

Khalidi, M. A. (2013). *Natural categories and human kinds: Classification in the natural and social sciences*. Cambridge University Press. https://doi.org/10.1017/CBO9781139026487

McGrew, K. S. (2009). CHC theory and the human cognitive abilities project: Standing on the shoulders of the giants of psychometric intelligence research. *Intelligence*, *37*(1), 1–10. https://doi.org/10.1016/j.intell.2008.08.006

Schneider, W. J., & McGrew, K. S. (2018). The Cattell-Horn-Carroll theory of cognitive abilities. In D. P. Flanagan & E. M. McDonough (Eds.), *Contemporary intellectual assessment: Theories, tests, and issues* (4th, pp. 73–163). The Guilford Press.