DNA: Distributed Neurocognitive Architecture

Cooperative AI Inference Without a Central Server Through Democratic Clusters and Verification

Centram Research | March 28, 2026 | Centram Whitepaper Series

Distributed Neurocognitive Architecture — how multiple AI nodes collaboratively generate and verify responses without any central server.

The Problem With Single-Model Inference

Today's AI operates in one of two modes. Either you run a model locally — constrained by your device's RAM, GPU, and the single perspective of one set of weights — or you send your query to a centralized API, trusting a corporation with your data, your context, and your reliance on their continued goodwill.

Both modes have fundamental limitations. A single local model has a ceiling on the quality it can produce. It has one set of weights, one training distribution, one perspective. When it hallucinates, there is nothing to catch it. When it encounters an out-of-distribution query, there is no fallback.

Centralized APIs solve the quality problem by running enormous models on dedicated hardware. But they introduce privacy risks, create single points of failure, and concentrate power in the hands of a few operators. When an API provider changes their terms of service, raises prices, or introduces content filters that don't align with your needs, you have no recourse.

DNA proposes a third path: **cooperative inference across multiple independent nodes**, where generation and verification are structurally separated, and no single node controls the process.

The Hub-Spoke Cluster Model

When a query enters the Centram network, it does not go to a single model. Instead, a **cluster** of nodes forms to handle it. Each cluster has one hub and multiple spoke nodes,

organized into two functional groups: generators and verifiers.

The hub is elected through a **two-thirds majority vote** among cluster members. This is a deliberate design choice. Simple majority (50%+1) is vulnerable to network partitions where two halves each elect a different hub. Two-thirds majority requires a genuine supermajority, making split-brain scenarios far less likely.

The election process works as follows: first, only nodes with a trust score above 0.6 are eligible as candidates. All cluster members then cast votes. If any candidate receives votes from at least two-thirds of the membership, they are elected hub. If no candidate reaches the threshold in the first round, a runoff is held among the top three vote-getters.

The hub coordinates the cluster but does not have unilateral authority. It assigns roles (generator or verifier), distributes the query, collects responses, and manages the verification loop. But it cannot override a verification failure or force acceptance of a response that verifiers have rejected.

Generation Clusters

Generator nodes receive the query and independently produce candidate responses. The key mechanism here is **lateral synchronization**: generators can exchange intermediate states during the generation process.

In a standard autoregressive model, each token is generated sequentially based on all previous tokens. In a DNA generation cluster, multiple generators work in parallel. At configurable synchronization points (typically every 32–64 tokens), generators broadcast their current hidden states to each other. Each generator then integrates the lateral signal with its own state before continuing.

Concretely, at every synchronization point (typically every 32 to 64 tokens), each generator broadcasts its current hidden state to the other generators in the cluster. It receives their states in return, computes a trust-weighted average of the peer signals, and blends this lateral signal into its own hidden state with a factor of 0.3. Then it continues generating from the blended state.

This produces responses that are **richer than any single node could generate alone**, because each generator benefits from the perspectives of its peers during the generation process. The blend factor of 0.3 is deliberately low — each node's own reasoning dominates, but lateral signals nudge it toward considering aspects it might have missed.

Verification Clusters

Once generators produce candidate responses, the verification cluster takes over. Verifiers are nodes that did **not** participate in generation — this structural separation is critical. A generator cannot verify its own output, just as a scientist cannot peer-review their own

paper.

Each verifier independently evaluates the candidate responses on multiple dimensions:

- **Factual consistency** — Do the claims in the response align with the verifier's own knowledge?
- **Logical coherence** — Does the reasoning chain hold together? Are there non-sequiturs or contradictions?
- **Completeness** — Does the response address all parts of the query?
- **Safety** — Does the response violate any of the verifier's configured content policies?

Verification results are submitted to the hub, which aggregates them. A response must achieve **2/3 verifier approval** to be delivered to the user. This threshold mirrors the hub election threshold — the two-thirds principle is applied consistently throughout DNA.

The Remand Loop

When verification fails — when a candidate response does not achieve the required approval threshold — the response is **remanded**. This is DNA's self-correction mechanism.

The remand loop operates as follows: the hub sends the query to generators, collects candidate responses, and forwards them to verifiers. If verification succeeds (two-thirds approval), the response is delivered. If it fails, the hub augments the original query with the specific rejection reasons from verifiers and sends it back to generators. This cycle repeats up to three times. If verification still fails after three cycles, the system applies graceful degradation — it delivers the best available response with an explicit low-confidence disclaimer, being transparent about the failure to reach consensus.

On remand, the hub sends the query back to the generators along with the **specific reasons for rejection**. Generators re-run inference with this additional context, producing revised candidates that address the verifiers' concerns. This cycle can repeat up to a configurable limit (default: 3 cycles) before the system applies graceful degradation — delivering the best available response with an explicit low-confidence disclaimer.

The remand loop is what gives DNA its self-correcting character. Unlike a single model that either gets it right on the first try or doesn't, DNA can iteratively refine its output through the dialectic between generation and verification.

Sovereignty Filtering

Every node in the Centram network is sovereign. This means each node controls what categories of queries it processes, what domains it participates in, and what content policies it enforces. A node operated by a medical professional might accept medical queries but refuse to participate in financial analysis clusters. A node operated by a parent

might refuse to process certain content categories entirely.

Sovereignty filtering happens at the cluster formation stage, before any inference begins. When the hub is assembling a cluster for a query, it only recruits nodes whose sovereignty policies permit participation. Nodes cannot be compelled to join a cluster, and they can withdraw at any time.

This is fundamentally different from centralized content moderation, where a single entity decides what is and isn't acceptable for everyone. In DNA, **each node makes its own decisions**, and the network functions as long as enough willing participants exist for a given query type.

Network Topology

DNA clusters don't form randomly. The underlying peer-to-peer network uses a **Small World topology** — a network structure where most nodes can reach most other nodes in a small number of hops, even though each node has relatively few direct connections.

Overlaid on the Small World structure is a **Rich-Club** pattern: highly connected, high-trust nodes tend to be connected to each other. This naturally emerges as nodes with good track records (high trust scores, high availability, high fitness genomes) are preferentially selected as cluster participants and therefore accumulate more connections.

The Cognitive Mesh Network consists of three overlay layers: the Evolution Network (Small World topology for genome exchange), the Verification Network (bipartite graph enforcing structural separation between generators and verifiers), and the Consensus Network (DAG structure for non-linear block confirmation). Nodes are classified by centrality: core nodes have high degree and eigenvector centrality, bridge nodes have high betweenness centrality (connecting otherwise separate clusters), and peripheral nodes are everything else.

The triple-overlay design separates concerns. The evolution network (IEIT layer) handles genome exchange. The verification network (DNA layer) enforces the structural separation between generators and verifiers using a bipartite graph. The consensus network handles blockchain confirmation using a DAG (Directed Acyclic Graph) structure for non-linear block ordering.

GWP Integration

DNA does not operate in isolation. It is the second layer of Centram's three-layer cognitive architecture. The outputs of DNA clusters — verified, consensus-approved responses — feed upward into the **Global Workspace Protocol (GWP)**.

GWP acts as a meta-cognitive layer. When multiple DNA clusters produce responses to related queries, or when a complex query requires synthesis across domains, GWP

collects the cluster outputs, detects contradictions between them, and issues deepening directives to resolve disagreements. The result is not just distributed inference, but distributed *reasoning* — a process where multiple perspectives are actively integrated rather than simply aggregated.

> *DNA provides the cooperative inference mechanism. It ensures that no query is answered by a single model operating in isolation, and that every response is independently verified before delivery. But cooperative inference alone is not enough. Verification catches errors; it does not synthesize insight. That is the role of the Global Workspace Protocol.*

Why Not Just Use a Bigger Model?

A reasonable objection to DNA is that you could achieve similar quality by simply running a larger model. If a 7B-parameter model hallucinates, run a 70B model. If that's not enough, run a 400B model.

This objection misses three things. First, **scale has diminishing returns**. The jump from 7B to 70B is dramatic; the jump from 70B to 700B is less so. Second, **larger models are not more diverse**. A 400B model trained on the same data distribution still has the same blind spots — just with higher confidence. Third, **larger models require centralization**. Only a handful of organizations can afford to run 400B+ models, which brings us back to the privacy and control problems that motivated DNA in the first place.

DNA achieves diversity through structural means. Multiple independent models, each evolved through IEIT to reflect different data distributions and user populations, collaborate and check each other's work. The result is a system that is **more robust than any single model**, regardless of that model's size, because the diversity is genuine — it comes from independent training histories, not from sampling the same model multiple times.

Centram © 2026 | Open Source under AGPL-3.0 | centram.org