

IEIT: Infective Evolutionary Intelligence Theory

How AI Models Evolve Across a Decentralized Network Through Peer-to-Peer Genome Exchange

Centram Research | March 28, 2026 | Centram Whitepaper Series

Infective Evolutionary Intelligence Theory — the biological metaphor that lets personal AI grow through peer-to-peer genome exchange.

When you train a large language model today, you need a data center. Thousands of GPUs, millions of dollars, months of compute. The result is a single monolithic model owned by a single corporation. IEIT proposes a fundamentally different path: what if AI models could **evolve** the way biological organisms do — through variation, selection, and inheritance across a distributed population?

Genomes: AI Weight Deltas as Heritable Material

In biology, a genome encodes the instructions for building an organism. In IEIT, a **genome** is a set of LoRA (Low-Rank Adaptation) weight deltas — the difference between a base model and a fine-tuned variant. These deltas are small (typically 10–50 MB compared to a multi-gigabyte base model), portable, and composable.

Each node on the Centram network runs its own local AI model. As the user interacts with it — asking questions, providing feedback, training on domain-specific tasks — the model accumulates adaptations stored as LoRA weight deltas. These deltas *are* the genome. They represent everything the local model has learned beyond its base training.

A genome is a structured package containing: a unique identifier, its lineage (the IDs of parent genomes it descended from), a generation number, a fitness score, the domain it specializes in (e.g. medical, legal, general), and the actual weight deltas organized by layer name. Every genome also carries an Ed25519 cryptographic signature from the node that produced it, ensuring that genomes cannot be forged or tampered with in transit.

P2P Genome Exchange

Genome exchange is the core mechanism of IEIT. When two nodes connect over Centram's P2P network, they can offer their best genomes to each other. This exchange follows a strict protocol:

1. **Discovery** — Nodes find peers via LAN UDP broadcast (port 9743) or known peer lists
2. **Authentication** — Mutual Ed25519 challenge-response handshake over the binary protocol
3. **Offer** — The sender advertises genome metadata (domain, fitness, size, generation)
4. **Accept/Reject** — The receiver applies sovereignty filters before accepting transfer
5. **Transfer** — For genomes larger than 32 MB, the GENOME_SHARD protocol splits the payload into chunks with individual acknowledgments
6. **Integration** — The receiver evaluates the genome and decides whether to incorporate it

For large genomes, the chunked transfer protocol (GENOME_SHARD message type 0x4E) splits payloads into 32 MB shards. The GenomeShardAssembler on the receiving end reconstructs the full genome and verifies integrity before passing it to the evaluation pipeline.

The 10-Layer Security Gate

Accepting foreign code into your AI model is inherently dangerous. A malicious genome could degrade performance, inject biases, or act as a backdoor. IEIT addresses this with a **10-layer security gate** that every incoming genome must pass through:

1. **Sovereignty Gate** — Does the genome's domain match categories this node accepts? Each node sets its own policy.
2. **Trust Gate** — Is the sender's trust score above the minimum threshold? Trust scores are computed from historical interaction quality.
3. **Rate Limit Gate** — Has this peer exceeded their genome submission quota? Prevents flooding attacks.
4. **Evaluation Budget Gate** — Does this node have enough compute budget remaining to evaluate the genome?
5. **Deserialization Gate** — Can the genome be safely deserialized without triggering code execution vulnerabilities?
6. **Structural Gate** — Do the weight delta shapes match the expected architecture? Rejects dimension mismatches.
7. **Magnitude Gate** — Are the weight values within expected bounds? Catches NaN, Inf, and absurdly large values.

8. **Statistical Gate** — Do the weight distributions look plausible? Detects adversarial patterns using Kolmogorov-Smirnov tests.

9. **Fitness Evaluation Gate** — Does the genome actually improve performance on local benchmarks?

10. **Fraud Detection Gate** — Does the genome exhibit signs of reward hacking or overfitting to evaluation metrics?

A genome must pass all 10 gates to be accepted. In practice, our experiments show a **41.9%** acceptance rate — meaning the gates are selective enough to filter out harmful genomes while permitting genuine improvements.

Differential Privacy

Even with the security gates, weight deltas can leak information about the training data that produced them. IEIT applies **per-layer adaptive differential privacy** to all outgoing genomes:

The process works as follows: for each layer, the weight deltas are first clipped to a maximum norm of 0.1, bounding their magnitude. Then, calibrated Gaussian noise is injected based on the privacy budget. The privacy parameters — epsilon of 2.0, delta of $1e-5$, and clip norm of 0.1 — strike a balance between privacy protection and model utility. At these settings, it is mathematically guaranteed that the presence or absence of any single training example cannot be reliably detected from the shared genome, while the weight deltas retain enough signal for meaningful evolutionary progress.

Tournament Selection, Crossover, and Blending

IEIT borrows three core mechanisms from evolutionary biology:

Tournament Selection (k=3)

When a node needs to select a parent genome for the next generation, it randomly samples **k=3** genomes from its local population and selects the one with the highest fitness. This provides moderate selection pressure — better genomes are more likely to reproduce, but outliers still get chances, maintaining diversity.

Crossover

Two parent genomes can be combined by taking different layers from each parent. For each layer in the model, the system randomly selects whether to inherit from parent A or parent B. This layer-level crossover preserves the internal coherence of each transformer layer while mixing high-level strategies across the full architecture.

Blending

Rather than discrete crossover, blending takes a weighted average of parent weights. The blend ratio can be uniform or fitness-proportional, giving more weight to the higher-performing parent.

FDAA: Fluctuation-Driven Adaptive Allocation

Not all domains evolve at the same rate. A node that primarily handles medical queries will accumulate medical-domain genomes faster than financial ones. FDAA models this as **activity waves** — each domain emits a signal proportional to its recent usage.

The key insight is the **basal metabolism floor**: even domains with zero recent activity maintain a minimum wave amplitude. This prevents cold-start death — a domain that falls silent can still receive incoming genomes and participate in evolution.

Wave synthesis combines local and remote signals at a 60/40 ratio: 60% weight on local domain activity, 40% on aggregated peer waves from the network. This combined signal is then modulated by two factors: a trust modifier (ranging from 0.7x to 1.3x based on peer trust scores) and a blend modifier (ranging from 0.9x to 1.3x based on genome diversity within the domain).

FDAA runs as a background task every 60 seconds, adjusting resource allocation across domains. Domains with strong wave activity get more compute budget for genome evaluation. Structural changes — promoting, demoting, splitting, or merging clusters — are triggered when sustained wave patterns indicate a shift in the network's cognitive landscape.

Minimum Viable Population

Evolution requires a minimum population size to function. Too few genomes and you get genetic drift — random noise overwhelms selection pressure. IEIT defines the **Minimum Viable Population (MVP)** as:

MVP is defined as the maximum of two values: $N_{\text{selection}}$ (tournament size plus one, typically 4) and $N_{\text{exploration}}$ (the inverse of the mutation rate). With tournament size $k=3$ and a typical mutation rate of 0.05, this gives $MVP = \max(4, 20) = 20$. A node needs at least 20 genomes in a domain before meaningful evolution can occur. Below this threshold, the system accumulates genomes through exchange without applying selection pressure.

Experimental Results

We validated IEIT across multi-node test deployments. Key findings:

- **Acceptance rate:** 41.9% of exchanged genomes pass all 10 security gates
- **Genome sizes:** Typically 10–50 MB for 7B-parameter models (LoRA rank 16–64)

- **Fitness convergence:** Population fitness stabilizes after approximately 30–50 generations in homogeneous domains
- **Privacy overhead:** Differential privacy noise adds less than 3% fitness degradation at $\epsilon=2.0$
- **Transfer throughput:** Chunked shard protocol sustains full bandwidth utilization for genomes up to 500 MB
- **Cross-platform compatibility:** Ed25519 signatures verified across Python (daemon), Swift (iOS), and Kotlin (Android) implementations

The 41.9% acceptance rate is significant. It means the security gates reject more than half of incoming genomes — protecting nodes from degradation — while still allowing enough genetic material through to sustain evolutionary progress.

The Biological Parallel

IEIT's name includes the word "infective" deliberately. In biology, horizontal gene transfer — where organisms acquire genes from non-parent organisms — is a powerful evolutionary mechanism. Bacteria exchange plasmids. Viruses insert DNA into host genomes. These "infections" can be harmful, but they also drive rapid adaptation.

IEIT formalizes this process for AI. A genome exchange is an *infection*: foreign genetic material enters the node's population. The 10-layer security gate acts as the immune system, filtering out pathogens while allowing beneficial genes through. Differential privacy ensures that the infection does not carry sensitive information from the source organism.

The result is a network where **AI models evolve continuously**, driven not by centralized training runs but by the collective pressure of thousands of nodes sharing, evaluating, and selecting genomes. Each node's AI becomes uniquely adapted to its owner's needs while benefiting from the collective intelligence of the entire network.

IEIT is the first layer of Centram's three-layer cognitive architecture. It provides the evolutionary substrate — the raw mechanism by which AI models improve over time. The second layer, DNA, governs how these evolved models cooperate during inference. The third layer, GWP, integrates their outputs into coherent, verified responses.