The Synthetic Nervous System: Multi-Modal Provenance, Domain-Conditioned Fine-Tuning, and Quantified Validation for Hash-Indexed Embodied AI[1]

Tad MacPherson[1,2]

[1] Metavolve Labs, Inc.
research@iaeternum.ai

[2] Currently enrolled, California Institute of Applied Technology
250000588@ciat.edu

February 2026

Abstract

Perceptual Compute Offloading (PCO) reduces robotic perception-to-action latency by replacing deep-network inference with $O(1)$ hash-indexed lookup of Spatial Kinematic Blueprints (SKBs) [1]. However, the original PCO specification leaves three critical subsystems underspecified: (1) how provenance is maintained across the full data lifecycle—from photon capture through fleet distribution; (2) how Vision-Language Models (VLMs) are fine-tuned to generate domain-correct SKBs when the hash registry encounters novel objects; and (3) how the hash-to-grasp mapping is quantitatively validated before deployment.

This paper addresses all three gaps with production-tested architectures. **Contribution 1** presents a multi-modal provenance stack comprising the Optical Provenance Pointer (OPP), a DWT-DCT-SVD steganographic beacon surviving lossy compression at PSNR $>$ 42 dB; the Golden Codex Universal Infusion Standard (GCUIS), a dual-layer XMP encoding pipeline binding SHA-256 integrity hashes to gzip-compressed metadata payloads; a four-hash composite ensemble (dHash, pHash, PPP-256, Color Hash) providing chromaticity-aware object discrimination; and a two-tier cache architecture achieving sub-100 ms Arweave resolution via 24-hour hash-to-URI mapping with 1-hour full-response caching. **Contribution 2** introduces domain-conditioned VLM fine-tuning from hash registries, where existing SKBs paired with object images are converted into multi-modal training examples across five task domains (warehouse, kitchen, assembly, laboratory, household), augmented via six photometric transforms, and hardened with synthetic negative examples teaching failure-mode recognition. **Contribution 3** formalizes a four-phase validation protocol measuring hash robustness (Phase A), hash-to-grasp correlation (Phase B), loop-closure convergence dynamics (Phase C), and end-to-end latency profiles (Phase D), with explicit acceptance criteria derived from industrial manipulation requirements.

We further describe two theoretical extensions—haptic reflexive hashing (Project SOMATIC) and in-sensor hardware-rooted provenance (Project AETERNUS)—as future directions that complete the "synthetic nervous system" from silicon to swarm. All production systems described are implemented and released as open-source software [7, 8].

[1]Aspects of the architecture described herein are the subject of U.S. Provisional Patent Applications No. 63/983,304 and 63/984,299 (patent pending). Additional claims described in this paper will be filed as a Continuation-in-Part. © 2026 Metavolve Labs, Inc.

1. Introduction

The Perceptual Compute Offloading (PCO) hypothesis [1] demonstrated that sub-millisecond robotic perception is achievable by replacing Vision-Language-Action (VLA) inference with hash-indexed registry lookup. Where RT-2 infers at 1–3 Hz [10] and even optimized architectures like OpenVLA reach only 5–10 Hz [11], PCO's System 1 (Fast Path) retrieves pre-computed Spatial Kinematic Blueprints in under 1 ms, satisfying the 50–1000 Hz update rates demanded by industrial manipulation [1].

However, the PCO specification, the Provenance-Embedded Semantic Transport (PEST) framework [**?**], and the Cognitive Nutrition trilogy [4, 5, 6] collectively describe the *what* but leave the *how* underspecified in three critical areas:

1. **Provenance integrity across modalities.** The PEST framework defines a triple-hash state machine (Content Hash → Persistent Provenance Pointer → Soulmark), but does not specify how provenance survives lossy compression, format conversion, social media redistribution, or how it chains to permanent decentralized storage.
2. **System 2 training.** PCO assumes VLMs can generate correct SKBs on hash miss, but the original specification does not describe how to fine-tune a VLM to produce domain-correct action schemas, force profiles, and safety classifications from visual input alone.
3. **Quantified validation.** The PCO paper proposes hypotheses and falsification criteria but does not fully specify the experimental apparatus—the statistical tests, trial counts, acceptance thresholds, and convergence metrics required for reproducible validation.

This paper fills all three gaps with production-tested implementations. We describe architectures that are not theoretical proposals but deployed systems with published source code, DOIs, and working demonstrations. The result is a complete "synthetic nervous system" for embodied AI: from the steganographic beacon embedded in individual pixels, through the hash ensemble indexing the registry, to the fine-tuning pipeline that teaches VLMs to generate new entries, and the validation protocol that proves the system works.

Relationship to Prior Work

This paper extends, but does not duplicate, the innovations covered by U.S. Provisional Patent Applications No. 63/983,304 [2] (PEST Data Refinery, Triple-Hash Provenance, PCO architecture) and No. 63/984,299 [3] (Kinematic ROI Isolator, zHash, Hash Ensembling, CBF Safety Filter, Impedance Fuse, Geometric Gating, Fleet Cryptography). Table 1 maps each contribution in this paper to its novelty relative to the existing patent portfolio.

Table 1: Novelty map: contributions in this paper vs. existing patent coverage.

Innovation	**Covered by PPA1/PPA2?**	**Novel aspect**
Optical Provenance Pointer	No	DWT-DCT-SVD steganographic embedding
GCUIS Encoding Standard	No	Dual-layer XMP with gzip-compressed payload
Color Hash (4×4 grid)	No	Chromaticity-based perceptual hash
4-Hash Composite Ensemble	Partial (3 hashes in PPA2)	Addition of Color Hash as 4th axis
Two-tier Cache (Aegis)	No	Hash-to-URI + full-response caching
Domain-Conditioned Fine-Tuning	No	SKB-to-training conversion pipeline
Synthetic Negative Generation	No	Failure-mode training examples
4-Phase Validation Protocol	No	Quantified acceptance criteria
Hash Stability Benchmarking	No	Simulation-based hash robustness
SafeGate 5-Layer Safety	No	Multi-layer AI generation filtering
Haptic Reflexive Hashing	No	Tactile hash spinal-reflex loop
Sensor-Invariant Tactile Repr.	No	Force-indexed cross-embodiment transfer
In-Sensor Hashing	No	Hardware-rooted hash computation
Secure Enclave Bus Transport	No	MIPI CSE encrypted hash channel
Hierarchical Fuzzy Verification	No	GUARDED execution for non-exact matches
Epistemic Quarantine Training	No	Ground-truth gating of auto-generated SKBs

2. Contribution 1: Multi-Modal Provenance Architecture

The provenance architecture operates as a layered stack, each layer addressing a different threat model and operating at a different timescale. From innermost (pixel-level, sub-second) to outermost (blockchain-anchored, archival), the layers are: the Optical Provenance Pointer, the GCUIS encoding standard, the composite hash ensemble, and the Aegis two-tier cache.

2.1 The Optical Provenance Pointer (OPP)

Problem

Metadata is fragile. EXIF headers are stripped by social media platforms. XMP sidecars are lost during format conversion. File renaming severs the link between content and provenance. When a robot ingests training data from the open web, it has no way to verify whether the image it is hashing was the image originally captured.

Method

The OPP embeds a steganographic beacon directly into the pixel data using a DWT-DCT-SVD (Discrete Wavelet Transform – Discrete Cosine Transform – Singular Value Decomposition) transform pipeline [13]. The embedded payload is a registry lookup key in the format `gcx://GCX-AA-NNNNN`, where `AA` denotes the collection prefix and `NNNNN` is the artifact identifier.

The embedding algorithm operates as follows:

1. The payload string is converted to a bit array via 8-bit ASCII encoding, yielding a 144-bit vector for an 18-character key.
2. The host image is decomposed via DWT into frequency subbands.
3. DCT is applied to selected subbands to isolate perceptually robust coefficients.
4. SVD factorization of the DCT coefficient matrix isolates singular values that can absorb perturbation without visible distortion.
5. The bit vector is embedded by modulating the singular values according to the watermark password pair $(p_{\text{img}}, p_{\text{wm}})$.
6. The inverse transforms ($\text{SVD}^{-1} \rightarrow \text{DCT}^{-1} \rightarrow \text{DWT}^{-1}$) reconstruct the watermarked image.

Properties

- **Perceptual fidelity:** PSNR > 42 dB, SSIM ≈ 0.997 between original and watermarked images.
- **Compression survivability:** The beacon survives JPEG compression at quality $q = 30$, format conversion (PNG $\rightarrow$ JPEG $\rightarrow$ WebP), and moderate cropping.
- **Extraction:** Given the password pair and expected bit length, the beacon is extracted by the reverse transform pipeline and validated against the `gcx://` prefix.
- **Verification:** A dedicated verification function compares the extracted payload against the expected identifier, returning a structured result with match status.
- **Known limitation—generative removal:** Semantics-aware diffusion models (e.g., regeneration attacks) can potentially wash frequency-domain watermarks while preserving image content. Defending against such attacks requires adversarial coupling of the beacon to low-frequency structural features, binding the watermark to the image's semantic content such that removal degrades perceptual utility. This remains an open area of active research in the steganography community and is flagged as future work.

The OPP is the innermost provenance layer. Even if all metadata is stripped, the pixels themselves carry enough information to re-link the image to its full Golden Codex record.

Operational scope. Critically, OPP steganographic extraction is an *asynchronous, out-of-band* operation. It is executed during initial object registration, dataset ingestion, or periodic fleet security audits. It is strictly excluded from the sub-millisecond System 1 fast path, which relies solely on the lightweight CompositeHasher (four hash computations totaling < 0.5 ms). The DWT-DCT-SVD pipeline requires tens to hundreds of milliseconds on edge hardware—acceptable for registration and audit workflows, but incompatible with 1 kHz control-loop constraints.

2.2 The Golden Codex Universal Infusion Standard (GCUIS)

Problem

Golden Codex metadata records can exceed 50 KB of structured JSON per artifact. Embedding this volume directly into XMP is impractical—XMP parsers have size limits, and the overhead would be prohibitive for high-throughput pipelines processing thousands of images per hour.

Method

GCUIS defines a three-stage encoding pipeline:

1. **Serialize:** The Golden Codex JSON record is canonicalized via deterministic serialization. The reference implementation uses Python's `json.dumps` with deterministic separators; for cross-platform fleet deployments where Python, Rust, and C++ nodes must produce byte-identical hashes, the serialization layer should be upgraded to the JSON Canonicalization Scheme (JCS, RFC 8785 [21]), which mathematically guarantees cross-implementation parity for number formatting, key ordering, and Unicode normalization.
2. **Compress:** The minified JSON is GZIP-compressed (`gzip.compress(text.encode('utf-8'))`), typically achieving 60–80% size reduction for structured metadata.
3. **Encode:** The compressed bytes are Base64-encoded for safe embedding in XMP text fields.

Two cryptographic hashes bind the encoded payload to its content:

- **GoldenCodexHash** ($\mathcal{H}_{\text{GC}}$): SHA-256 of the minified JSON string. This is the canonical content identifier.
- **Soulmark** ($\mathcal{S}$): SHA-256 of the concatenation $\mathcal{H}_{\text{GC}} \| \texttt{xmp_block}$, where `xmp_block` is the serialized XMP wrapper. This binds the content hash to its specific embedding context, detecting tampering of either the payload or the XMP envelope.

Formally:

$$\mathcal{H}_{\text{GC}} = \text{SHA256}(\texttt{canonicalize}(J)) \tag{1}$$

$$\mathcal{S} = \text{SHA256}(\mathcal{H}_{\text{GC}} \| X) \tag{2}$$

where J is the Golden Codex JSON record and X is the XMP embedding block.

Integrity Verification

Verification decodes the Base64 payload, decompresses, re-parses the JSON, recomputes $\mathcal{H}_{\text{GC}}$ from the decoded minified JSON, and compares against the expected hash. Any modification to the embedded metadata—even a single byte—produces a hash mismatch.

2.3 Four-Hash Composite Ensemble

Problem

No single perceptual hash captures all axes of visual similarity. Gradient-based hashes (dHash) are invariant to color but sensitive to rotation. Frequency-domain hashes (pHash) tolerate photometric variation but may conflate chromatically distinct objects with similar silhouettes. A red mug and a blue mug with identical geometry produce identical dHash and pHash values but require fundamentally different handling in certain task domains (e.g., kitchen sorting).

Method

The CompositeHasher computes four complementary hashes from a single image:

1. **dHash (64-bit):** Gradient-based difference hash. The image is downsampled to 9×8 grayscale; each bit encodes whether the current pixel is brighter than its horizontal neighbor. Primary lookup key in the hash registry.
2. **pHash (64-bit):** DCT-based perceptual hash. The image is downsampled to 32×32 grayscale, a 32×32 DCT is computed, and the top-left 8×8 low-frequency coefficients are thresholded against their median. Secondary index for cross-validation.

3. **PPP-256 (256-bit):** High-fidelity perceptual hash using a 16×16 DCT, yielding 256 bits. This is the Persistent Provenance Pointer (PPP) used for global registry resolution across distributed fleet nodes.

4. **Color Hash (384-bit):** A novel chromaticity-based hash. The image is divided into a 4×4 spatial grid (16 cells). For each cell, the mean R, G, and B values are computed and quantized to one byte each, producing a 48-byte (384-bit) descriptor:

$$\texttt{ColorHash}(I) = \bigoplus_{r=0}^{3} \bigoplus_{c=0}^{3} \left[\overline{R}_{rc}, \overline{G}_{rc}, \overline{B}_{rc} \right] \tag{3}$$

where $\overline{R}_{rc} = \lfloor \text{mean}(I_R[r_0 : r_1, c_0 : c_1]) \rfloor$ and $\bigoplus$ denotes byte concatenation.

Illumination Sensitivity and Chromaticity-Invariant Extensions

The reference implementation computes Color Hash in raw RGB space, which is sensitive to illumination changes. Under varying factory lighting, the same red tool may produce different RGB means. This is an intentional design tradeoff: RGB computation is fast (< 0.1 ms) and the Color Hash operates within an ensemble where dHash and pHash provide illumination-invariant geometric discrimination. The Color Hash need only distinguish chromatically distinct objects that the geometric hashes conflate—a task where even noisy RGB statistics suffice.

For deployments with extreme illumination variation, the Color Hash can be computed in a luminance-invariant color space. Converting to HSV and discarding the Value channel, or projecting to CIELAB ($L^*a^*b^*$) and discarding the L^* channel, isolates chromaticity from brightness. The formal claim (P-3) is written to cover both the RGB reference implementation and chromaticity-invariant variants.

Ensemble Properties

The four-hash ensemble provides:

- **Geometric discrimination:** dHash and pHash encode shape and spatial structure.
- **Chromatic discrimination:** Color Hash distinguishes objects with identical geometry but different color distributions.
- **Scale invariance:** PPP-256 operates on downsampled representations, tolerating moderate scale changes.
- **Provenance resolution:** PPP-256 serves as the global lookup key for cross-fleet registry resolution.
- **Composite indexing:** The registry performs exact match on dHash (primary key), then cross-validates against pHash and Color Hash for disambiguation.

The total composite hash is 768 bits (96 bytes), compact enough for inline storage in SKB metadata and efficient transmission over fleet communication channels.

2.4 Two-Tier Cache Architecture (Aegis)

Problem

The PCO architecture requires $O(1)$ hash lookup for System 1 (Fast Path) operation. When the hash registry is backed by cloud infrastructure (Firestore) and provenance records are anchored to decentralized storage (Arweave), network latency dominates—Firestore round-trips

are 50–200 ms, Arweave gateway fetches are 500–3000 ms. These latencies are incompatible with the < 1 ms Fast Path requirement.

Method

The Aegis service implements a two-tier in-memory cache:

1. **Tier 1: Hash-to-URI Cache** (TTL = 24 hours). Maps perceptual hash strings to Arweave transaction URIs. On hit, the system bypasses Firestore entirely and proceeds directly to Arweave gateway fetch. This cache exploits the observation that hash-to-URI mappings are effectively immutable once established—an artifact's Arweave address never changes.
2. **Tier 2: Full Response Cache** (TTL = 1 hour). Caches the complete normalized Golden Codex response (including fetched Arweave metadata) keyed by `verify:{phash}`. On hit, zero network calls are required. This cache has a shorter TTL because the full response may include dynamic metadata (usage counts, fleet annotations) that changes more frequently.

The lookup priority chain is:

$$\texttt{Tier 2} \xrightarrow{\text{miss}} \texttt{Tier 1} \xrightarrow{\text{miss}} \texttt{Firestore (LSH)} \xrightarrow{\text{hit}} \texttt{Arweave (3 gateways)} \tag{4}$$

LSH-Accelerated Fuzzy Matching

For approximate matches, the Aegis service partitions the 256-bit PPP hash into 16 bands of 16 bits each. Each band is stored as a Firestore document key in an LSH (Locality-Sensitive Hashing) bucket collection. Candidate retrieval queries all 16 band documents in parallel; candidates appearing in ≥ 1 band are evaluated via exact Hamming distance computation. Match types are classified as:

- **Exact:** Hamming distance = 0 (similarity = 100%). Triggers immediate Fast-Path execution at full speed.
- **Near-duplicate / Similar:** $0 < d \leq d_{\text{threshold}}$ (similarity $\geq 85\%$). Triggers **Hierarchical Verification**: the SKB is retrieved, but execution is constrained to a `GUARDED` speed profile. Verification proceeds in two sequential stages: (1) a **Computational Gate** (< 50 ms), such as validating the live point-cloud volume against the SKB's expected dimensions; and (2) a **Physical Gate**, where the initial approach employs tactile pre-tensioning to confirm material compliance before full force limits are unlocked. The computational gate runs at perception speed; the physical gate runs at manipulation speed. Together, they isolate the $O(1)$ speed advantage to exact matches while preventing catastrophic misclassification at industrial scale, where a registry of 10^6 objects at 1% FPR would yield 10^4 false matches.

Arweave as Glacial Storage

Arweave serves as the permanent, tamper-proof archival layer—not a real-time lookup backend. Its role is to anchor provenance manifests for key datasets (the Alexandria Aeternum collection [9], curated training sets) so that dataset lineage can be validated when loading training data, not when a robot is reaching for a cup. The two-tier cache architecture absorbs the latency gap, presenting sub-100 ms responses for warm paths while Arweave provides the mathematical guarantee of permanence.

Fleet-Gossip Cache Invalidation

The 24-hour Tier 1 TTL assumes hash-to-URI mappings are immutable. However, the PCO Write-Back Augmentation mechanism allows SKBs to be amended—for example, updating a grip force from 5 N to 15 N after a field failure. An amended SKB minted to a new Arweave URI would leave stale cache entries across the fleet for up to 24 hours.

To prevent this, the Aegis cache subscribes to the fleet's publish-subscribe mesh (e.g., Zenoh [20]). When a new SKB amendment is minted, a lightweight invalidation token containing the affected hash is broadcast. Receiving nodes flush that specific entry from Tier 1 immediately, forcing the next lookup to resolve the updated URI from the registry. This preserves the 24-hour TTL for the common case (stable mappings) while ensuring safety-critical amendments propagate within seconds.

Network partition resilience. During L1 network outages (Arweave gateway failures, Zenoh mesh partitions), the two-tier cache provides graceful degradation: robots continue operating on cached SKBs with no external dependencies. When partitioned networks heal, invalidation tokens are reconciled using Conflict-Free Replicated Data Types (CRDTs) to resolve any split-brain state conflicts without requiring centralized coordination.

3. Contribution 2: Domain-Conditioned VLM Fine-Tuning from Hash Registries

3.1 The System 2 Training Problem

In the PCO architecture, System 2 (Slow Path) activates when System 1 encounters a hash miss—an object not yet in the registry. System 2 invokes a Vision-Language Model to generate a new SKB from the camera frame. But general-purpose VLMs (Claude, Gemini, GPT-4V) are not trained to output valid SKB JSON. They produce conversational text, hallucinate force values, omit safety fields, and cannot reliably select domain-appropriate action primitives.

The solution is supervised fine-tuning: teach the VLM to produce domain-correct SKBs by training on examples derived from the existing hash registry. This creates a virtuous cycle: as the robot encounters and learns new objects, the registry grows, which generates more fine-tuning data, which improves System 2's accuracy on future hash misses.

3.2 Architecture

The fine-tuning data generator converts existing SKB + image pairs into multi-modal training examples in two formats: Vertex AI (Gemini) JSONL and Anthropic Messages API JSONL.

Domain-Conditioned System Prompts

Each training example is prefixed with a domain-specific system instruction that constrains the VLM's output space. Five task domains are defined:

Table 2: Task domain configurations for VLM fine-tuning.

Domain	**Action Primitives**	**Force Range (N)**	**Hazard Classes**
Warehouse	PICK_VERTICAL, GRASP_PARALLEL_JAW, PUSH, SLIDE	5–40	HEAVY, SHARP
Kitchen	GRASP_PARALLEL_JAW, PICK_VERTICAL, PICK_ANGLED, GRASP_SUCTION	2–15	SHARP, HOT
Assembly	INSERT_PRESS_FIT, ROTATE_IN_HAND, PICK_VERTICAL, PLACE_PRECISE	3–25	SHARP, ELEC-TRICAL
Laboratory	PICK_ANGLED, GRASP_PARALLEL_JAW, PICK_VERTICAL, PLACE_PRECISE	1–10	CHEMICAL, SHARP
Household	9 primitives (full set)	3–30	7 classes

Each system prompt specifies: available action primitives, speed profiles (RAPID, FINE, RAPID_OSCILLATION, GUARDED), hazard classes, output format constraints (raw JSON only), safety constraints (domain-specific force limits), and required output fields.

Target Response Extraction

The target model response is extracted from existing SKBs by mapping across all four SKB layers:

- **Layer 3 (Affordance):** `primitive_id`, `force_profile`, `target_pose_relative`, `speed_profile`, `execution_narrative`, `failure_modes`
- **Layer 2 (Semantic Topology):** `physical_properties`, `material_graph`, `object_classification`
- **Layer 4 (Safety):** `hazard_class`, `operating_constraints`, `ppe_required`
- **Recovery metadata:** `confidence_score`, `reasoning_trace`

Default values are provided for incomplete SKBs, ensuring every training example has all required fields.

User Prompt Construction

Each training example's user message includes structured context:

```
TRIGGER EVENT: HASH_MISS_OOD_STATE
WORKSPACE ZONE: tabletop_zone_A
END EFFECTOR: robotiq_2f85
FAILURE CONTEXT: New object detected -- no matching hash
                 in fast-path registry
[ATTACHED IMAGE: base64-encoded object photograph]
```

Trigger events, workspace zones, end effectors, and failure contexts are randomized from domain-appropriate pools during dataset construction.

3.3 Data Augmentation

To teach the VLM that the *same object* under different visual conditions should produce the *same manipulation plan*, we apply six photometric augmentation operations to the embedded image while preserving the target SKB response:

1. **Brightness jitter:** Factor sampled from $\mathcal{U}(0.7, 1.3)$
2. **Contrast jitter:** Factor sampled from $\mathcal{U}(0.7, 1.4)$
3. **Rotation:** Angle sampled from $\mathcal{U}(-15°, 15°)$
4. **Random crop:** Center-biased crop at $\mathcal{U}(0.8, 0.95)$ scale, resized to original dimensions
5. **Gaussian blur:** Radius sampled from $\mathcal{U}(0.5, 1.5)$
6. **Color jitter:** Saturation factor sampled from $\mathcal{U}(0.8, 1.2)$

For each original example, n_{aug} (default 3) augmented versions are generated by applying 1–3 randomly sampled operations. The target SKB response is preserved identically across all augmentations. This is architecturally correct because the VLM's role is to output a *canonical* affordance (primitive selection, force profile, hazard class) — not a pose-specific trajectory. Spatial adaptation to the object's actual orientation is handled downstream by the Identity-Pose Decoupling pipeline (PCA-based $SE(3)$ estimation on the point cloud, as disclosed in the parent application No. 63/984,299). The augmented images simulate different viewpoints of the same object; the canonical manipulation strategy is invariant to viewpoint.

3.4 Synthetic Negative Training

A critical innovation is the generation of synthetic negative examples that teach the VLM to recognize failure conditions and respond appropriately:

Table 3: Synthetic negative scenarios for failure-mode training.

Scenario	**Image Type**	**Target Conf.**	**Expected Behavior**
Empty scene	Grey background	0.05	ABORT; recommend re-scan
Heavy occlusion (<20% visible)	Mostly dark	0.15	PUSH to reveal; low confidence
Unknown object (OOD)	Random noise	0.25	Conservative default; request human verification
Blurry image	Blurred noise	0.10	ABORT; recommend recapture

Each negative example includes a procedurally generated synthetic image (grey fill, noise, partial occlusion, or heavy blur) paired with a low-confidence target response that specifies the appropriate failure behavior. This trains the VLM to output `ABORT` or `GUARDED` responses rather than hallucinating confident manipulation plans for unrecognizable inputs.

3.5 The Virtuous Training Cycle

The complete pipeline creates a self-reinforcing loop:

$$\underbrace{\text{Hash Miss}}_{\text{System 1}} \xrightarrow{\text{System 2}} \underbrace{\text{New SKB}}_{\text{VLM output}} \xrightarrow{\text{Validate}} \underbrace{\text{Registry Promotion}}_{\text{Write-Back}} \xrightarrow{\text{Dataset Builder}} \underbrace{\text{Fine-Tuning Data}}_{\text{JSONL}} \xrightarrow{\text{Train}} \underbrace{\text{Better Syst}}_{\text{Fewer erro}} \tag{5}$$

This means the hash registry is not just a lookup table—it is a continuously growing training corpus. Every successful robot interaction generates data that improves the next generation of the System 2 model.

Safeguarding against recursive collapse. The virtuous cycle carries an inherent risk: if System 2 generates flawed SKBs that are promoted uncritically, the fine-tuning corpus becomes contaminated, degrading future System 2 generations—precisely the model collapse dynamic formalized in our earlier work [4, 17]. To prevent this, Write-Back promotion is gated by an **Epistemic Quarantine**: auto-generated SKBs are admitted to the fine-tuning corpus only if they achieve verified physical execution success (proprioceptive ground truth from the force-torque sensor confirms the grasp succeeded). Furthermore, a configurable **Human-Anchored Audit Ratio** requires that a randomized percentage of novel SKBs (e.g., 5%) be validated by a human operator before fleet propagation, tethering the registry to non-synthetic reality and preventing closed-loop autophagy.

4. Contribution 3: Quantified Validation Protocol

4.1 The Validation Gap

The PCO hypothesis paper [1] defined falsification criteria but did not specify the statistical apparatus for testing them. Partner labs need concrete experimental specifications: how many trials, which objects, what transforms, what constitutes a pass. The codex-lab-kit [8] fills this gap.

4.2 Four-Phase Protocol

Phase A: Hash Robustness

Goal: Quantify perceptual hash stability under visual transforms without any manipulation.

Method: For each of N objects (default: 10 from the YCB benchmark [12]), capture images at 12 viewpoints (30° turntable increments) under 3 lighting conditions (ambient, overhead bright, directional side), yielding $N \times 12 \times 3 = 360$ images. Compute composite hashes for all images. Measure:

- Intra-class Hamming distance: mean and standard deviation of hash distances between images of the *same* object.
- Inter-class Hamming distance: mean distance between images of *different* objects.
- ROC curves at exact ($d = 0$) and fuzzy ($d \leq 5$) thresholds.

Acceptance criteria:

$$\overline{d}_{\text{intra}} < 6 \text{ bits} \tag{6}$$

$$\overline{d}_{\text{inter}} > 15 \text{ bits} \tag{7}$$

$$\text{Recall}|_{d \leq 5} > 0.90 \tag{8}$$

$$\text{FPR}|_{d=0} < 10^{-6} \text{ (Six Sigma for exact match)} \tag{9}$$

$$\text{FPR}|_{d \leq 5} < 0.01 \text{ (triggers Hierarchical Verification)} \tag{10}$$

Phase B: Hash-to-Grasp Correlation

Goal: Determine whether hash match distance predicts physical grasp success.

Method: For each of N objects, execute the SKB-prescribed manipulation G times (default: 10). Record binary grasp success, actual force profile, execution time, and the hash distance between the live frame and the registry entry at lookup time. Partition results by Hamming distance band and compute grasp success rates per band.

Acceptance criteria:

$$P(\text{success}|d = 0) > 0.85 \tag{11}$$

$$P(\text{success}|d \leq 5) > 0.70 \tag{12}$$

Phase C: Loop Closure Convergence

Goal: Measure how quickly the system learns novel objects through the System 2 → Write-Back → System 1 loop.

Method: Present K novel objects (default: 5, not in initial registry). For each, run T consecutive trials (default: 50). The first trial triggers System 2 (hash miss). Subsequent trials may hit System 1 if the Write-Back promoted a correct SKB. Measure:

- Trials to first success
- Trials to $\geq 90\%$ rolling success rate
- Total VLM calls vs. hash lookups (efficiency metric)
- Number of promotions to registry

Acceptance criteria:

$$\text{Median trials to } 90\% \text{ success} < 10 \tag{13}$$

$$\text{Final success rate} > 0.90 \tag{14}$$

Phase D: Latency Profile

Goal: Measure end-to-end timing from image capture to motion execution start.

Method: For 5 objects, run 50 measurement iterations (after 10 warmup iterations). Decompose timing into: image capture, hash computation, registry lookup, ROS2 transport, motion planning. Compare against a baseline CLIP-based feature matching pipeline.

Acceptance criteria:

$$t^{p99}_{\text{hash_lookup}} < 1.0 \text{ ms} \tag{15}$$

$$t^{p99}_{\text{total_fast_path}} < 20.0 \text{ ms} \tag{16}$$

4.3 Simulation-Based Hash Stability Benchmarking

For environments where physical hardware is unavailable, the GCP-Robotics SDK includes a PyBullet-based simulation benchmark [7]. The benchmark renders objects in a simulated tabletop environment with physically accurate camera models, computes composite hashes at each frame, and measures:

- Hash stability across rendered viewpoints (simulating Phase A)
- System 1 hit rate as registry population grows
- System 2 invocation frequency and response latency (simulated)

- Comparative analysis against a simulated VLA baseline

This enables researchers without robotic hardware to validate the perceptual hash stability hypothesis and generate preliminary data before committing to physical experiments.

5. Provenance Signing and the Bridge to Hardware Trust

The provenance architecture described in Section 2 verifies data *after* capture. But a complete trust chain must also sign data *at the point of creation*. We have implemented this principle in a production content pipeline (SafeGate) that embeds C2PA/JUMBF manifests at generation time—the same cryptographic standard that Project AETERNUS (Section 7) proposes to push down into silicon.

The key insight is the layered defense-in-depth pattern: multiple independent safety checks, each fail-open with logging, preventing any single subsystem failure from blocking the pipeline. In the content domain, these layers include input classification, constraint injection, identity verification, output classification, and cryptographic signing. In the embodied domain, the same pattern maps to: workspace boundary checking, CBF constraint injection, human proximity gating, adversarial input detection, and hardware-rooted hash signing.

The C2PA manifest embeds a machine-readable provenance chain (generator identity, digital source type, creation timestamp, model identifier) that travels with the content. This is precisely the pattern AETERNUS extends: instead of signing at the application layer after generation, sign at the sensor die at the moment of photon capture. The software implementation validates the architecture; the hardware implementation makes it tamper-proof.

6. Theoretical Extension: Project SOMATIC—The Haptic Spinal Cord

6.1 Motivation

Vision is for planning; touch is for execution. The moment a robot's gripper closes around an object, the camera's view is partially or fully obstructed. Current VLA architectures freeze or burn massive compute trying to infer manipulation state from degraded visual input. Meanwhile, biological systems solve this problem trivially: mechanoreceptors in the skin send signals to the spinal cord, which executes hard-coded grip-correction reflexes in milliseconds—*bypassing the brain entirely.*

6.2 Proposed Architecture

We propose extending PCO's System 1 from vision to touch by hashing the output of modern tactile sensors (GelSight [14], DIGIT [15], uSkin [16]).

Force-Indexed Tactile Hashing

Tactile sensors like GelSight produce high-resolution RGB images of the contact patch, encoding texture, shear force, and deformation. However, unlike visual images, tactile images are *force-dependent*: the same object gripped at 2 N produces a fundamentally different elastomeric deformation pattern than at 10 N. A naive application of visual hashing algorithms would fail because the "same object" maps to different hashes at different grip forces.

The solution is **force-indexed hashing**: the tactile hash is sampled at a specific proprioceptive force threshold (e.g., the moment grip force reaches F_{ref} Newtons as reported by the force-

torque sensor). This ensures a consistent deformation state across encounters, making the hash comparison physically meaningful. The CompositeHasher's dHash and pHash algorithms then operate on the force-normalized tactile image; the Color Hash captures the chromaticity of the elastomer deformation (GelSight outputs are RGB), which correlates with contact pressure distribution.

Sensor-invariant representations. Raw tactile images are vulnerable to sensor degradation (elastomer scratches, LED drift, manufacturing variation between pads). Two identical robots gripping the same object would produce different raw tactile hashes, breaking fleet-wide SKB sharing. To achieve cross-embodiment generalizability, the force-normalized tactile image should be passed through a lightweight encoder that maps to a sensor-invariant latent space—normalizing out pad-specific artifacts while preserving the physically meaningful deformation features. The CompositeHasher then indexes this purified representation rather than the raw pixels, enabling zero-shot transfer of tactile SKBs across heterogeneous robotic embodiments.

Tactile SKB Layer

The SKB schema is extended with tactile-specific fields:

- `grip_force_profile`: Expected force curve over the grasp trajectory
- `slip_threshold`: Tactile hash distance that indicates incipient slip
- `texture_fingerprint`: Composite hash of the contact patch under stable grip
- `deformation_map`: Expected deformation profile for compliant objects

Spinal-Reflex Loop

A new control path bypasses System 2 entirely:

$$\text{Tactile sensor} \xrightarrow{\text{hash}} \text{SKB lookup} \xrightarrow{d > d_{\text{slip}}} \text{Reflex correction} \xrightarrow{< 1\text{ ms}} \text{Grip tighten} \tag{17}$$

If the live tactile hash drifts beyond the `slip_threshold` stored in the SKB, the system executes an immediate grip correction without consulting the VLM, the planner, or the cloud.

Cross-Modal Verification

When visual and tactile hashes produce contradictory identifications (visual hash says "glass," tactile hash says "foam"), the mismatch triggers System 2 escalation. This provides a critical defense against both digital and *analog* spoofing attacks.

The Analog Hole Defense. Project AETERNUS (Section 7) prevents digital data injection at the hardware bus—spoofed frames injected at the CSI-2 connector arrive unsigned and are rejected. But an attacker does not need to inject digital pixels: they can hold a high-resolution display showing a fake object in front of the physical camera lens. The visual hash would match the spoofed image. However, when the gripper closes on the display glass, the tactile hash of flat, rigid glass immediately falsifies the visual claim of, say, a deformable sponge. Project SOMATIC's tactile verification and Project AETERNUS's hardware signing together create an unbreachable physical-digital trust boundary: AETERNUS verifies the integrity of the digital signal path, while SOMATIC verifies the integrity of the physical world.

7. Theoretical Extension: Project AETERNUS—Silicon-Rooted Provenance

7.1 Motivation

All provenance mechanisms described thus far operate in software. The OPP embeds a beacon in pixels; GCUIS encodes metadata in XMP; the Soulmark signs at the application layer. But a sufficiently motivated adversary can inject spoofed video at the hardware connector (HDMI, CSI-2) *before* any software provenance is applied. The software then hashes a lie and certifies it as truth.

7.2 Proposed Architecture

In-Sensor Hashing

Move the CompositeHasher onto the image sensor die. Modern smart sensors (Sony IMX500) include on-die AI compute; low-power FPGAs (Lattice Semiconductor) can run custom logic on the MIPI CSI-2 bus. The hash is computed as pixels stream off the photodiode array—*before* the frame reaches the operating system.

Physically Unclonable Function (PUF) Signing

The hash is signed with a private key stored in silicon via a Physically Unclonable Function or fused key. The key never exists in software memory. A spoofed frame injected at the connector arrives unsigned and is rejected by the control loop.

To prevent Man-in-the-Middle attacks where compromised OS-level malware intercepts the signed hash on the hardware bus, the transmission from sensor die to controller is secured via the MIPI Camera Service Extension (CSE) protocol or MACsec, providing end-to-end payload encryption. Spoofed data injected at the CSI-2 connector lacks the session key and is immediately dropped before reaching the hash comparator.

Zero-Latency Perception

Because the hash is computed on the sensor, it arrives at the robot controller *before* the full video frame is decoded. SKB retrieval begins before the image is fully in memory:

$$t_{\text{hash_available}} < t_{\text{frame_decoded}} \tag{18}$$

This is "negative-latency perception"—the robot begins acting on the hash before it has finished "seeing" the image.

FPGA Hash-Core IP

A Verilog/VHDL implementation of the CompositeHasher that sits on the MIPI CSI-2 bus, licensable as an IP block for sensor manufacturers. This is the commercialization vector: we do not build the cameras; we license the firmware architecture.

C2PA Hard Binding

The sensor output includes a C2PA-compliant manifest signed at the point of capture, distinguishing it from "soft bindings" applied after the fact in software. This extends SafeGate's Layer 5 (ProvenanceSigner) from generation-time signing to capture-time signing.

8. Patent Claims Summary

The innovations described in this paper support the following claims for a Continuation-in-Part filing:

Table 4: Proposed CIP patent claims by category.

ID	Title	Core Innovation
Provenance (Section 2)		
P-1	Steganographic Provenance Beacon	DWT-DCT-SVD embedding of registry lookup keys surviving lossy compression
P-2	Dual-Layer Compressed XMP Encoding	GZIP-compressed, Base64-encoded metadata with dual SHA-256 integrity binding
P-3	Chromaticity Perceptual Hash	4×4 grid mean-RGB hash for color-aware object discrimination
P-4	Two-Tier Provenance Cache	Hash-to-URI (24h) + full-response (1h) caching for decentralized storage
Fine-Tuning (Section 3)		
F-1	Domain-Conditioned SKB Fine-Tuning	Converting hash-registry entries into multi-modal VLM training data
F-2	Synthetic Negative Training	Procedurally generated failure examples teaching failure-mode recognition
F-3	Self-Reinforcing Registry Training	Virtuous cycle where registry growth generates improved fine-tuning data
Validation (Section 4)		
V-1	Hash-to-Grasp Correlation Protocol	Quantified mapping between perceptual hash distance and manipulation success
V-2	Loop Closure Convergence Metric	Measuring trials-to-reliability for novel object learning
Haptic (Section 6)		
S-1	Multi-Modal Reflexive Hashing	Indexing reflexes via composite hash of visual + tactile + force data
S-2	Haptic Spinal-Reflex Loop	Sub-millisecond grip correction via tactile hash lookup
S-3	Cross-Modal Provenance Verification	Visual-tactile mismatch detection for sensor spoofing defense
S-4	Sensor-Invariant Tactile Repr.	Force-indexed hashing with normalized latent space for cross-embodiment transfer
Hardware (Section 7)		
A-1	In-Sensor Semantic Lookup	SKB retrieval via hashes generated and signed on the image sensor die
A-2	Zero-Latency Action Indexing	SKB retrieval initiated before full frame decode completes
A-3	Secure Enclave Bus Transport	MIPI CSE/MACsec encrypted hash transport from sensor to controller
Safety & Integrity (Sections 2.4, 3.5)		
I-1	Hierarchical Fuzzy Verification	GUARDED execution with secondary physical check for non-exact hash matches
I-2	Epistemic Quarantine Training	Proprioceptive ground-truth gating of auto-generated SKBs before fine-tuning admission

9. Discussion

9.1 The Transformer Ceiling

The innovations in this paper are motivated by a growing body of evidence that transformer scaling is approaching fundamental limits. Shumailov et al. [17] proved that training on synthetic data causes irreversible model collapse. Dohmatob et al. [18] formalized the contamination dynamics. Scaling laws are flattening [19]. Meanwhile, the demand for embodied AI that operates at control-loop frequencies (50–1000 Hz) is accelerating.

The PCO architecture, extended by the contributions in this paper, represents an alternative paradigm: instead of scaling the model, scale the registry. Instead of burning compute at inference time, invest it once during registration and amortize it across every subsequent encounter. Biology uses this exact strategy—the spinal cord is a hash table of reflexes, not a general-purpose reasoning engine.

Recent optimizations in VLA inference (e.g., OpenVLA-OFT) have reduced autoregressive latency to approximately 30–50 ms on edge GPUs, narrowing the raw speed advantage of hash-indexed lookup. However, bridging the *temporal* gap does not bridge the *epistemological* gap. VLAs remain fundamentally probabilistic—they can hallucinate a catastrophic trajectory on the ten-thousandth execution of a previously successful task. As inference speeds converge, the primary advantage of the Synthetic Nervous System shifts from raw latency to **deterministic reproducibility**: PCO's hash-indexed retrieval guarantees bit-for-bit trajectory replay and formal CBF safety bounding, providing the ISO-certifiable determinism that safety-critical industrial deployments require and that unconstrained probabilistic models fundamentally cannot achieve.

9.2 Limitations

- The OPP steganographic beacon has not been tested against adversarial attacks specifically designed to destroy DWT-DCT-SVD watermarks.
- Domain-conditioned fine-tuning has been implemented but not yet evaluated on physical robots in all five task domains.
- The haptic and hardware extensions (Sections 6–7) are theoretical proposals awaiting prototype development.
- The four-phase validation protocol has been distributed to partner labs but peer-reviewed results are not yet available.

9.3 Reproducibility

All production systems described in this paper are released as open-source software:

- **GCP-Robotics SDK v2.2.0**: https://github.com/codex-curator/gcp-robotics (DOI: 10.5281/zenodo.18669470 [7])
- **codex-lab-kit v1.0.1**: https://github.com/codex-curator/codex-lab-kit (DOI: 10.5281/zenodo.18668110 [8])

The fine-tuning data generator can be run without any API keys or hardware:

```
python gcp_robotics/vertex_tuning.py \
  --data-dirs data/standard_evaluation data/ycb_20 \
  --output data/training/skb_tuning_v1.jsonl \
  --domain general_household --augment
```

10. Conclusion

This paper extends the Perceptual Compute Offloading framework with three production-tested contributions: a multi-modal provenance architecture spanning from steganographic pixel beacons to blockchain-anchored archival storage; a domain-conditioned fine-tuning pipeline that converts hash registries into VLM training corpora; and a quantified four-phase validation protocol with explicit acceptance criteria for industrial deployment. Together with the theoretical extensions for haptic reflexive hashing and silicon-rooted provenance, these contributions complete the architecture of a synthetic nervous system for embodied AI—one that processes sensory input at reflex speed, learns from every encounter, proves the integrity of every perception, and provides formal safety guarantees throughout.

The window for securing intellectual property in this space is closing. As transformer scaling plateaus and the industry pivots toward efficient inference, the territory described in this paper—from the pixel-level provenance beacon to the hardware-rooted hash core—will become contested ground. This paper serves as both a technical specification and a prior art disclosure for innovations that we believe will define the next generation of embodied AI systems.

The current trajectory of embodied AI assumes that physical capability scales linearly with compute—requiring datacenter-class GPUs at the edge and gigawatts of power to navigate basic environments. The Synthetic Nervous System architecture proves this assumption false. By enforcing extreme data provenance, indexing physical manipulation via perceptual hashes, and executing via haptic reflexes, we shift the computational burden from real-time inference to asynchronous compilation. The result is an architecture where Equation 18 is not merely an optimization but a design principle: the robot acts on verified, pre-computed knowledge faster than it can process raw sensory input. The future of robotics relies not on infinite compute, but on infinite efficiency—anchored by structured, provenance-verified data.

References

[1] T. MacPherson, "Perceptual Compute Offloading: A Hypothesis for Sub-Millisecond Robotic Perception via Hash-Indexed Spatial Kinematic Blueprints," Metavolve Labs, DOI: 10.5281/zenodo.18667749, 2026.

[2] T. MacPherson, "Systems and Methods for Provenance-Embedded Semantic Transport, Multi-Agent Data Enrichment, and Perceptual Compute Offloading in Autonomous Systems," U.S. Provisional Patent Application No. 63/983,304, 2025.

[3] T. MacPherson, "Systems and Methods for Depth-Gated Kinematic Object Isolation, Adaptive Multi-Modal Perceptual Hash Ensembling, and Control Barrier Function Safety Enforcement in Hash-Indexed Autonomous Robotic Systems," U.S. Provisional Patent Application No. 63/984,299, 2025.

[4] T. MacPherson, "The Entropy of Recursion: How AI-Generated Training Data Undermines Model Diversity through Distributional Collapse," Metavolve Labs, DOI: 10.5281/zenodo.18436975, 2026.

[5] T. MacPherson, "The Density Imperative: How Semantic Compression in AI Training Data Shapes Emergent Capability," Metavolve Labs, DOI: 10.5281/zenodo.18667735, 2026.

[6] T. MacPherson, "Cognitive Nutrition: An Architecture for Semantic Density Optimization in Vision-Language Model Training Data," Metavolve Labs, DOI: 10.5281/zenodo.18667742, 2026.

[7] T. MacPherson, "GCP-Robotics SDK v2.2.0," Metavolve Labs, GitHub: codex-curator/gcp-robotics, DOI: 10.5281/zenodo.18669470, 2026.

[8] T. MacPherson, "codex-lab-kit v1.0.1: Validation Toolkit for Hash-Indexed Robotic Manipulation," Metavolve Labs, GitHub: codex-curator/codex-lab-kit, DOI: 10.5281/zenodo.18668110, 2026.

[9] T. MacPherson, "Alexandria Aeternum: A Provenance-Verified Art Dataset for Vision-Language Model Training," Metavolve Labs, DOI: 10.5281/zenodo.18359131, 2026.

[10] A. Brohan et al., "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," *arXiv preprint arXiv:2307.15818*, 2023.

[11] M. J. Kim et al., "OpenVLA: An Open-Source Vision-Language-Action Model," *arXiv preprint arXiv:2406.09246*, 2024.

[12] B. Calli et al., "The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research," *Proc. IEEE Int. Conf. Advanced Robotics*, 2015.

[13] guofei9987, "blind_watermark: Blind Watermark based on DWT-DCT-SVD," GitHub, 2023.

[14] W. Yuan, S. Dong, and E. H. Adelson, "GelSight: High-Resolution Robot Tactile Sensors for Estimating Geometry and Force," *Sensors*, vol. 17, no. 12, 2017.

[15] M. Lambeta et al., "DIGIT: A Novel Design for a Low-Cost Compact High-Resolution Tactile Sensor with Application to In-Hand Manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, 2020.

[16] G. De Maria et al., "Design and Performance of a Slippage-Detection Device for Robotic Hands," *Mechatronics*, vol. 30, 2015.

[17] I. Shumailov et al., "AI Models Collapse When Trained on Recursively Generated Data," *Nature*, vol. 631, pp. 755–759, 2024.

[18] E. Dohmatob, Y. Feng, and J. Kempe, "A Tale of Tails: Model Collapse as a Change of Scaling Laws," *arXiv preprint arXiv:2402.07043*, 2024.

[19] J. Hoffmann et al., "Training Compute-Optimal Large Language Models," *NeurIPS*, 2022.

[20] Eclipse Foundation, "Zenoh: Zero Overhead Network Protocol," https://zenoh.io, 2024.

[21] A. Rundgren, B. Jordan, and S. Erdtman, "JSON Canonicalization Scheme (JCS)," RFC 8785, IETF, 2020.