Perceptual Compute Offloading:

A Hypothesis for Sub-Millisecond Robotic Perception via Hash-Indexed Spatial Kinematic Blueprints[1][2]

Tad MacPherson[1,2]

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

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

February 2026

Abstract

Vision-Language-Action (VLA) models represent a breakthrough in robotic generalization, enabling robots to ground semantic instructions directly into physical actions using internet-scale knowledge. However, state-of-the-art VLA architectures impose a latency tax that renders them incompatible with high-frequency robotic control: RT-2 infers at 1–3 Hz, Octo at 2.5 Hz, and even optimized architectures like OpenVLA achieve only 5–10 Hz—while industrial manipulation demands 50–1000 Hz update rates. We term this the **Inference-Control Discrepancy**.

We hypothesize that **Perceptual Compute Offloading** (PCO) via a dual-process architecture can reduce robotic perception-to-action latency by $\geq$90% for known environmental states while preserving VLA-level generalization for novel encounters. System 1 (Fast Path) isolates the target object via depth-gated kinematic ROI extraction, computes a composite perceptual hash, and retrieves a pre-computed **Spatial Kinematic Blueprint** (SKB)—a four-layer action schema encoding provenance, physical properties, manipulation primitives, and safety constraints—from a hash-indexed registry in sub-millisecond time. System 2 (Slow Path) activates only on hash miss or low-confidence states, invoking full VLA inference to generate new SKBs that are promoted to the registry via **Write-Back Augmentation**, which pre-computes hashes for geometric, photometric, and occlusion variations to ensure robustness. Safety is enforced through **Control Barrier Functions** derived directly from SKB constraints, providing formal guarantees that perception errors are contained by deterministic physical envelopes.

This architecture is grounded in the **Provenance-Embedded Semantic Transport** (PEST) framework, empirically validated in the Cognitive Nutrition trilogy where the same hash-to-metadata principle produced +160% semantic coverage improvement in vision-language tasks. We present formal hypotheses with explicit falsification criteria and a standardized four-phase experimental protocol distributed as the open-source `codex_lab_kit` [34]. This paper is an invitation: the first partner labs to validate or falsify these hypotheses will be co-authors on the results publication.

1. Introduction

1.1 The Inference-Control Discrepancy

Robotic manipulation and locomotion are governed by the physics of contact and inertia. To maintain stability during force-sensitive tasks—peg insertion, surface wiping, surgical instrument handoff—a controller must update actuator commands faster than the timescale of the system's

[1] Aspects of the architecture described herein are the subject of U.S. Provisional Patent Applications (patent pending). © 2026 Metavolve Labs, Inc.

[2] DOI: 10.5281/zenodo.18667749

natural dynamics. Contact-rich manipulation typically requires 500–1000 Hz control loops, while even high-level trajectory planners need 10–50 Hz update rates to respond to dynamic obstacles [1].

Vision-Language-Action models solve a different problem: generalization. By mapping images and language instructions to action sequences through internet-scale pre-training, VLAs enable robots to handle novel objects and instructions without task-specific programming. This capability comes at prohibitive computational cost. A single VLA forward pass requires encoding high-resolution visual data, fusing it with language tokens, and autoregressively decoding action tokens—a process that scales linearly with model depth and sequence length.

The result is a fundamental structural mismatch. We define the **Inference-Control Discrepancy** as the ratio between the control frequency required by the robotic task (f_{control}) and the inference frequency achievable by the VLA (f_{VLA}):

$$\text{ICD} = \frac{f_{\text{control}}}{f_{\text{VLA}}} \quad \text{where ICD} \gg 1 \text{ for current architectures} \tag{1}$$

For RT-2 performing contact-rich manipulation, ICD $\approx$ 333–1000×. Even optimized architectures like π_0-FAST (50–200 Hz) achieve ICD $\approx$ 1–20×, and do so by reintroducing architectural bifurcation—separating a fast execution path from the slow semantic backbone. This architectural trend is not a coincidence. It is convergent evolution toward dual-process cognition.

1.2 The Cognitive Nutrition Connection

The same framework that enriches cultural heritage artifacts with dense semantic metadata can enrich robotic perception. In the art domain, the Provenance-Embedded Semantic Transport (PEST) protocol maps an image to a 256-bit Persistent Provenance Pointer (PPP) via perceptual hash, which indexes a 111-field semantic metadata record (NEST) encoding composition, technique, cultural context, affect, and symbolism [14, 16].

The Cognitive Nutrition experiment [15] proved that this dense structured metadata dramatically improves vision-language model capabilities: +160% semantic coverage, +8.9% CogBench, and a 63-point swing between sparse and dense training conditions on the same model architecture. Critically, sparse metadata *destroyed* pre-trained capabilities (−54.4% CogBench, +330% hallucination rate), establishing that data quality is *signed*—it either teaches or lobotomizes.

We propose that the same principle applies to robotic manipulation. The mapping becomes:

Art domain:	Image → PPP hash → 111-field semantic metadata
Robotics domain:	Camera frame → PPP hash → Spatial Kinematic Blueprint

If dense structured metadata produces measurably better model behavior in vision-language tasks, dense structured action metadata—indexed by the same perceptual hash—should similarly outperform real-time VLA inference for known environmental states.

1.3 Contributions

This paper makes eight contributions:

1. **Formal specification** of a dual-process architecture for robotic perception that bridges the Inference-Control Discrepancy
2. **The Spatial Kinematic Blueprint** (SKB): a four-layer schema (Provenance, Semantic Topology, Affordance, Safety) encoding complete manipulation knowledge with SKB versioning for kinematic refinement

3. **The Kinematic ROI Isolator**: a zero-CNN depth-gated pipeline for sub-millisecond object isolation, ensuring System 1 contains no neural network inference
4. **Write-Back Augmentation**: addressing geometric invariance, end-effector occlusion, and specular variation through pre-computed augmented hash keys
5. **Object-Conditioned Hash Ensembling**: inverse-variance weighted composite distance that autonomously learns which hashes to trust per object
6. **Control Barrier Function safety**: formal mathematical guarantees that perception errors are contained by deterministic physical envelopes
7. **A standardized four-phase experimental protocol** with codified acceptance criteria, distributed as the open-source `codex_lab_kit`
8. **Explicit falsification criteria** for all hypotheses, ensuring rigorous evaluation

2. Background and the Inference-Control Discrepancy

2.1 The VLA Latency Landscape

Table 1 synthesizes latency benchmarks across the current generation of VLA architectures, categorized by generation and suitability for real-time control.

Table 1: VLA inference latency benchmarks (2024–2026). Control requirements: contact-rich manipulation 500–1000 Hz; trajectory planning 10–50 Hz.

Model	**Architecture**	**Latency**	**Max Freq.**	**Suitability**
RT-2 [2]	Autoregressive (PaLI-X)	330–1000 ms	1–3 Hz	Low
OpenVLA [3]	Autoregressive (Llama-2)	100–200 ms	5–10 Hz	Low
Octo [4]	Diffusion Transformer	∼400 ms	2.5 Hz	Low
Xiaomi R-0 [5]	VLM + DiT	80 ms	30 Hz	Medium
π_0-FAST [6]	Compressed tokens	5–20 ms	50–200 Hz	High
HyperVLA [7]	Hypernetwork	∼4 ms	>100 Hz	High

Generation 1 architectures (RT-2, OpenVLA, Octo) demonstrate remarkable generalization but operate at 1–10 Hz, far below the threshold for real-time closed-loop control. Every action requires a complete forward pass through a large vision-language model, creating a "compute wall" that prevents direct servoing.

Generation 2 architectures (π_0-FAST, HyperVLA, Xiaomi Robotics-0) begin to close the gap, but they do so by *reintroducing architectural bifurcation*. π_0-FAST compresses action tokens into temporally extended trajectory chunks, trading fine-grained reactivity for speed. HyperVLA uses a hypernetwork to generate small, task-specific policies on the fly, avoiding activation of the full VLA for every inference step. Xiaomi Robotics-0 pipelines a frozen VLM for semantic understanding with a specialized Diffusion Transformer for action generation.

The convergent insight: every system that achieves real-time performance does so by **decoupling semantic reasoning from sensorimotor execution**. This is precisely the dual-process architecture we formalize. However, as we argue in Section 8, even Generation 2 architectures

carry fundamental limitations—in power consumption, determinism, and fleet editability—that PCO resolves by a different mechanism entirely.

2.2 The Action Chunking Compromise

To bridge the gap between 3 Hz inference and 500 Hz control, nearly all current VLA implementations rely on **action chunking**: a single inference step predicts a sequence of future actions (e.g., the next 50 waypoints), which a low-level controller executes while the VLA computes the next chunk [8].

While effective for slow, quasi-static tasks, action chunking has critical failure modes in dynamic environments:

1. **Open-loop vulnerability**: Once a chunk is dispatched, the robot is blind to new sensory information until the chunk completes. If an obstacle moves into the path during the 500 ms execution window, the robot collides.
2. **Jerky transitions**: Blending consecutive chunks produces velocity/acceleration discontinuities, reducing precision and increasing actuator wear [5].
3. **Temporal inconsistency**: Discretized action tokens are too coarse for tasks requiring continuous high-frequency feedback—threading a needle, pouring liquid, or modulating contact force [6].

Action chunking is a necessary compromise, not a solution. It addresses symptoms of the Inference-Control Discrepancy without resolving its structural cause.

2.3 Convergent Evidence for Dual-Process Architectures

Dual-Process Theory, formalized by Kahneman and Stanovich [9, 10], posits that human cognition operates through two complementary systems: System 1 (fast, automatic, pattern-matched) and System 2 (slow, deliberative, analytical). The analogy to robotic control is not superficial. Human experts rely on pattern-matched rapid responses for routine tasks (catching a ball, driving a familiar route), reserving deliberative reasoning for novel situations (navigating an unfamiliar city, diagnosing a new mechanical failure). The human brain accounts for 20% of metabolic energy despite being 2% of body weight [27]—a purely System 2 robot would be similarly paralyzed by its own energy budget.

Recent robotics literature (2024–2026) provides striking convergent validation:

AdaDrive [11] implements a "Slow-Fast" paradigm for autonomous driving. A lightweight trajectory planner (System 1) operates on every frame at >30 Hz, while a large Vision-Language Model (System 2) activates at low frequency to process long-range context. Connector modules fuse the LLM's high-level guidance into the fast planner's cost function.

AdaptiX [12] introduces **adaptive threshold switching** in assistive robotics. A meta-layer dynamically adjusts the sensitivity of the system based on environmental complexity and user behavior: when the environment is simple, the rule-based engine (System 1) is favored; when complexity increases, the CNN-based engine (System 2) is engaged. This directly validates the need for dynamic τ in our architecture.

MeCo [13] applies **memoization** to multi-robot LLM-based planning. It identifies similar sub-tasks across missions and retrieves previously computed plans instead of re-querying the LLM. A similarity testing method validates whether cached plans apply to the current context—functionally identical to our perceptual hash trigger. Experiments demonstrate substantial

latency and cost reductions with improved success rates due to the deterministic nature of reused plans.

These independent convergences confirm that dual-process cognition is the correct architectural abstraction for integrating foundation models into robotic control. Our contribution is the specific bridging mechanism: a **hash-indexed registry of Spatial Kinematic Blueprints**.

2.4 The PEST Framework and Cognitive Nutrition Validation

The Provenance-Embedded Semantic Transport framework [14] provides the architectural foundation for PCO. Its key components:

- **Persistent Provenance Pointer (PPP)**: A 256-bit perceptual hash embedded in or computed from media, serving as a content-addressed lookup key
- **Triple-Hash Protocol**: Content hash (SHA-256) for bitwise integrity $\oplus$ perceptual hash for similarity-based retrieval $\oplus$ soulmark for content-addressed identity, creating a four-state provenance machine (Pristine / Transformed / Tampered / Unresolvable)
- **Four-Layer Metadata Ontology**: Structured schemas that bind rich semantic context to perceptual identifiers

The Cognitive Nutrition experiment [15] validated this framework empirically using the Alexandria Aeternum dataset [33]. Three conditions—Base (no fine-tuning), Group A (sparse metadata, $\sim$50 tokens/image), and Group B (dense NEST, $\sim$2,000–4,000 tokens/image)—were applied to Llama 3.2 11B Vision-Instruct on 9,081 cultural heritage images, holding architecture and hyperparameters constant:

Metric	Base	Group A	Group B	B vs Base
CogBench Overall	0.384	0.175	**0.418**	+8.9%
Semantic Coverage	0.099	0.028	**0.257**	+160.3%
Hallucination Rate	**0.010**	0.043	0.013	+33%

The 63-point CogBench swing between sparse and dense conditions establishes that dense structured metadata teaches models *how to think about* a domain, not merely *what to say about* specific examples. We hypothesize that the same principle applies when the domain shifts from cultural heritage to robotic manipulation: dense structured action metadata (SKBs) will produce superior manipulation performance compared to real-time VLA inference for known states.

3. Hypotheses

We formalize three testable hypotheses:

H1 (Primary — Latency Reduction): For a robotic manipulation task operating within a Template Environment with a pre-populated SKB registry, a dual-process architecture using composite perceptual hash lookup (System 1) with VLA fallback (System 2) will achieve equivalent or superior task success rates compared to continuous VLA inference, while reducing mean perception-to-action latency by at least 90%.

H2 (Secondary — Convergence): The System 1 fast path will achieve >90% visual recognition hit rate after a single Write-Back Augmentation cycle for a novel object, with full kinematic convergence (grasp parameter refinement via proprioceptive feedback) within 8 physical trials.

H3 (Safety): Hash-indexed action execution will not introduce novel failure modes beyond those present in the underlying VLA system, provided: (a) the adaptive hash threshold τ is

calibrated to the application domain, (b) object-conditioned hash ensembling is employed, and (c) SKB safety constraints are enforced as Control Barrier Functions at the RTOS level.

Null Hypothesis: The architecture fails to deliver meaningful benefits if any of the following hold:

1. False Acceptance Rate exceeds 0.1% in manipulation trials (System 1 executes wrong SKB at industrially unacceptable frequency)
2. Total fast-path perception-to-action latency exceeds 10 ms at p_{99} (insufficient advantage over Generation 2 VLAs)
3. SKB-indexed grasp success degrades by >2% relative to the VLA baseline that generated the SKB (caching mechanism is lossy)
4. Kinematic convergence requires >8 physical trials to reach 90% grasp success on novel objects (learning dynamics too slow)

4. Proposed Architecture

4.1 System Overview

The Perceptual Compute Offloading architecture operates as a collaborative loop between two asynchronous processes:

1. **System 1 (Fast Path):** Camera frame → Kinematic ROI Isolator (Section 4.3) → Composite hash of isolated crop → Registry query with object-conditioned weighted distance (Section 4.6) → SKB retrieval → Secondary verification → CBF-constrained action chunk execution. Target: <10 ms end-to-end (100 Hz) for chunk retrieval, feeding a 1 kHz RTOS joint-impedance execution loop. **No neural network inference in the critical path.**
2. **System 2 (Slow Path):** Triggered on hash miss, low confidence, proprioceptive exception, or validation failure → Full VLA inference → SKB generation → **Write-Back with Augmentation** (Section 4.5) → Registry promotion → System 1 resumes. Latency: 200–2000 ms (acceptable because it occurs only for novel states).
3. **Safety Hold:** During System 2 deliberation, System 1 executes a safety hold—maintaining the current pose with compliant impedance control, or executing a pre-defined safe retreat. The SKB's Control Barrier Functions (Section 4.8) remain active throughout.

The "surprise" signal that transfers control from System 1 to System 2 is the hash match outcome. A match within threshold τ indicates a familiar state (low surprise). A miss or high Hamming distance indicates novelty (high surprise). This mirrors the prediction-error mechanism that governs attentional switching in biological dual-process systems [9]. Additionally, **proprioceptive exceptions**—force/torque anomalies detected during execution (e.g., object slip, unexpected contact)—trigger System 2 independently of hash state, enabling kinematic parameter refinement (Section 4.9).

4.2 Spatial Kinematic Blueprints

The Spatial Kinematic Blueprint is the canonical artifact of the architecture—a four-layer schema encoding the complete manipulation knowledge for a single object/action-primitive pair. The schema is implemented as a Pydantic v2 model aligned with the patent FIG 2 reference architecture (U.S. Provisional Application No. 63/983,304).

Layer 1 — Provenance (ref 200): Origin, calibration, and environmental context. Manufacturer, model, serial number, calibration date, workspace zone, lighting condition, enrichment timestamp and model version.

Layer 2 — Semantic Topology / NEST (ref 210): Object classification (7 types: Rigid_Body_Graspable, Deformable, Articulated, Liquid_Container, Fragile, Tool, Fastener), physical properties (mass, center of mass, friction coefficient, dimensions), material graph, kinematic joints for articulated objects, and the multi-hash perceptual fingerprint for fast-path lookup—including per-hash reliability weights (Section 4.6).

Layer 3 — Affordance (ref 220): Action schema with 13 canonical primitives (PICK_VERTICAL through HANDOVER), SE(3) target pose *relative to the visual centroid of the ROI mask* (position + quaternion), force profile (grip force, torque limits, compliance stiffness), speed profile (RAPID, FINE, RAPID_OSCILLATION, GUARDED), validation signature (expected post-action hash with tolerance and failure recovery), and an ordered execution narrative of trajectory acts.

Layer 4 — Safety (ref 230): Hazard classification (7 classes), PPE requirements, regulatory certifications (ISO 10218-1, CE), operating constraints (max speed, max force, restricted zones parameterizing Control Barrier Functions), enterprise licensing, and export restrictions.

The registry stores **trajectory chunks**, not single action steps. Each SKB's affordance layer contains an ordered execution narrative of 10–20 waypoints or spline segments, aligned with the action chunking paradigm validated by π_0-FAST [6]. This ensures that System 1 execution is smooth and continuous. Crucially, unlike standard VLA action chunking which executes open-loop in the world frame (Section 2.2), PCO chunks are **Continuously Anchored**. Because the Kinematic ROI Isolator (Section 4.3) tracks the target's visual centroid at >500 Hz, the executing $SE(3)$ waypoints are dynamically translated in real-time. If the object shifts during the execution window, the trajectory shifts with it, transforming a static chunk into a closed-loop tracking behavior.

Worked example. Consider a bin-picking cell where Part A becomes mechanically wedged against Part B. System 1's hash lookup fails (HASH_MISS_OOD_STATE). System 2 activates, assesses the entanglement via VLA inference (1835 ms), and generates a new SKB with primitive SHAKE_BIN_HORIZONTAL: lateral oscillation at 15 Nm torque limit, RAPID_OSCILLATION speed profile, with a validation signature that expects a specific post-action hash (tolerance 5 bits). On success, the post-action hash matches, and the SKB is promoted to the fast path with 20+ augmented hash keys. On the next entanglement, System 1 resolves it in 0.8 ms.

4.3 The Kinematic ROI Isolator

A perceptual hash of the full camera frame would be destroyed by irrelevant background changes—a forklift crossing the factory floor, a coworker walking past, or the robot's own arm entering frame during approach. System 1 therefore **never hashes raw frames**. It hashes an isolated, depth-gated crop of the target object.

The Kinematic ROI Isolator achieves this in sub-millisecond time **without neural network inference**:

Step 1 — Kinematic Self-Masking (<0.1 ms): The robot knows its own joint configuration. Using the URDF (Unified Robot Description Format) and forward kinematics, the system projects the robot's physical volume into the camera frame and masks it out, eliminating end-effector occlusion noise.

Step 2 — Depth-Plane Subtraction (<0.2 ms): Using the RGB-D camera's hardware-accelerated depth map, a strict Z-axis cutoff removes the static background (table surface, walls, floor). Only foreground objects within the workspace depth envelope remain.

Step 3 — Fast Clustering and Tracking (<0.5 ms): Connected Component Labeling (CCL) on the foreground point cloud identifies distinct objects. The bounding box centroid is wrapped in a Kalman filter with sparse Lucas-Kanade optical flow for temporal coherence. Full depth-CCL is re-executed only when the optical flow prediction error exceeds a threshold, amortizing the computation across frames.

The resulting ROI crop is passed to the CompositeHasher. The hash is therefore invariant to background factory clutter, robot arm position, and global scene changes. The ROI's 3D centroid anchors the SKB's `target_pose_relative`: when System 1 matches a hash, it applies the SKB's relative SE(3) transform to the *current* centroid position, bridging the scale-invariant hash to the physical world.

Critically, PCO explicitly **decouples semantic identity from continuous spatial configuration**. The perceptual hash provides discrete *semantic identity*—retrieving the canonical SKB affordance ("this is a 10mm hex bolt; grasp with 15N, approach vertically"). The isolated 3D point cloud provides *continuous kinematic anchoring*—tracking the object's $SO(3)$ orientation via PCA eigenvectors or lightweight ICP alignment. System 1 dynamically applies a continuous $SE(3)$ affine transform to the retrieved SKB trajectory based on the live point cloud pose. This prevents the quantization errors typical of discrete template matching (e.g., a 4° rotation causing a step-function grasp error), enabling sub-degree precision for contact-rich assembly without the combinatorial explosion of exhaustive $SE(3)$ augmentation.

4.4 The Hash Engine

The CompositeHasher computes four perceptual representations of each ROI crop, each capturing a different perceptual axis:

- **dHash (64-bit):** Gradient-based difference hash. Captures structural layout of edges. Primary lookup key for fast-path retrieval.
- **pHash (64-bit):** DCT-based perceptual hash. Captures low-frequency structural "bones" of the image. Secondary index for cross-validation.
- **pHash-256 (256-bit):** High-fidelity perceptual hash used as the PPP for global registry resolution across distributed nodes.
- **Color Hash (4×4 grid):** Mean RGB per 16 grid cells. Distinguishes geometrically identical but chromatically distinct objects (e.g., a red mug vs. a blue mug with identical dHash).
- **zHash (64-bit):** Depth hash computed on the normalized 2.5D depth map from the RGB-D sensor. Because depth encodes intrinsic surface geometry rather than reflected light, zHash is *mathematically invariant* to ambient lighting, shadows, and specular glare. On machined metal parts where sliding specular highlights destroy RGB gradient hashes, Inverse-Variance Weighting (Section 4.6) autonomously shifts matching weight to zHash.

For sub-millisecond retrieval at scale, the registry is indexed via a **BK-Tree** or **Multi-Index Hashing** (MIH) table, which support Hamming-distance nearest-neighbor queries without linear scan. At 1M registered objects, MIH achieves sub-millisecond lookup [17].

Honest assessment of limitations. Standard pHash and dHash are robust to high-frequency noise, JPEG compression, and global illumination changes [18]. However, they are *not rotation invariant*. Because DCT basis functions are direction-dependent, rotating an image by even 5–10 degrees significantly alters the hash coefficients [19]. Additionally, gradient-based hashes (dHash) are vulnerable to **specular glare** on machined metal or glossy plastic surfaces, where a small viewpoint change violently shifts highlight positions, destroying local gradients. We address geometric invariance through Write-Back Augmentation (Section 4.5), specular robustness

through Object-Conditioned Hash Ensembling (Section 4.6), and environmental optical control (Section 4.7).

4.5 Write-Back Augmentation

Write-Back Augmentation addresses the geometric and photometric invariance weaknesses of perceptual hashing. When System 2 generates a new SKB for ROI crop $\mathbf{I}_t$, the hash engine *also* computes hashes for a comprehensive set of augmentations:

$$\mathcal{A}(\mathbf{I}_t) = \{T(\mathbf{I}_t) \mid T \in \{R_{\pm\theta}, S_{\pm\sigma}, L_{\pm\lambda}, O_k, G_j\}\} \tag{2}$$

where $R_{\pm\theta}$ are in-plane rotations at $\pm 5°, \pm 10°, \pm 15°$; $S_{\pm\sigma}$ are scale variations at $\pm 10\%, \pm 20\%$ (simulating camera distance changes); $L_{\pm\lambda}$ are brightness, contrast, and color temperature perturbations; O_k are **random erasing masks and synthetic gripper silhouettes** simulating end-effector occlusion during the approach phase; and G_j are **localized specular perturbations** simulating glint shifts on reflective surfaces.

For environments requiring out-of-plane viewpoint robustness, System 2 can optionally employ lightweight depth-aware view synthesis (e.g., depth heuristic reprojection) to generate hash keys for pitch/yaw variations that 2D homographic warping cannot accurately simulate.

All augmented hashes point to the *same SKB*:

$$\forall T \in \mathcal{A} : \text{Registry}[\text{Hash}(T(\mathbf{I}_t))] \rightarrow \text{SKB}_t \tag{3}$$

This moves the cost of robustness from **runtime** (where it would add latency to every System 1 query) to **background processing** (where it occurs once during System 2 write-back). For a typical augmentation set of 30 transforms, the additional write-back cost is ∼150 ms—negligible compared to the 200–2000 ms VLA inference that precedes it. Storage cost is minimal: 30 augmented hash keys add only 240 bytes per object.

During the augmentation phase, the system also computes the **per-hash intra-class variance** (σ_k^2) across the augmentation set for each hash type k, which is stored in the SKB and used for Object-Conditioned Hash Ensembling (Section 4.6).

4.6 Object-Conditioned Hash Ensembling

A static voting scheme for multi-hash consensus (e.g., "3 of 4 hashes must match") is brittle. On a machined steel gear, specular glare shifts will break dHash and Color Hash simultaneously under slight viewpoint changes, while pHash (low-frequency DCT structure) remains stable. On a matte cardboard box, all four hashes are equally reliable. The optimal weighting depends on the specific object and its material properties.

We replace static voting with **inverse-variance weighted composite distance**. During Write-Back Augmentation (Section 4.5), the system computes the intra-class variance σ_k^2 of each hash type $k \in \{\text{dHash}, \text{pHash}, \text{Color}, \text{pHash-256}, \text{zHash}\}$ across the augmentation set. The normalized weight for each hash type is:

$$w_k = \frac{1/(\sigma_k^2 + \epsilon)}{\sum_j 1/(\sigma_j^2 + \epsilon)} \tag{4}$$

These weights are stored directly in the SKB's perceptual signature. At runtime, System 1 computes the weighted composite distance:

$$D_{\text{comp}} = \sum_k w_k \cdot \left(\frac{\Delta_k}{N_k} \right) \tag{5}$$

where Δ_k is the Hamming distance for hash type k and N_k is the bit-length of that hash type, normalizing all distances to a uniform $[0, 1]$ fractional scale. Without this normalization, the 256-bit pHash-256 ($\Delta_{\max} = 256$) would mathematically dominate the 64-bit dHash ($\Delta_{\max} = 64$) regardless of the variance weights. A match is accepted when $D_{\text{comp}} < \tau(t)$.

The architecture *autonomously teaches itself* to ignore specular glare by down-weighting fragile hashes on a per-object basis. No manual tuning is required. The weights are calibrated automatically during the Write-Back phase for every object in the registry.

Furthermore, because Write-Back Augmentation (Section 4.5) injects synthetic gripper silhouettes (O_k) into the augmentation set, the inverse-variance weighting mathematically learns to down-weight the spatial frequencies associated with the end-effector's approach path. When the Kinematic Self-Mask (Section 4.3) replaces the robot arm with black pixels during final approach, these "black hole" artifacts are already represented in the per-hash variance estimates. The system autonomously learns to ignore masking artifacts during the terminal grasp phase—precisely when hash stability is most critical.

4.7 Template Environments and Optical Engineering

A **Template Environment** is a fully pre-hashed workspace where every object has a registered SKB with a multi-view template database and calibrated hash weights. A new robot drops into a Template Environment and immediately operates at System 1 speed with zero cold start.

Target domains:

- **Assembly lines**: Every part, tool, fixture, and bin. Each component pre-hashed from 12+ viewpoints under 3+ lighting conditions.
- **Commercial kitchens**: Every utensil, appliance, and ingredient container. SKBs encode fragility, pour angles, and thermal constraints.
- **Biopharma labs**: Every pipette, centrifuge, and reagent bottle. SKBs include contamination avoidance constraints and sterile handling protocols.
- **Hospital ORs**: Every surgical instrument, supply cart, and monitor. Safety Layer 4 includes regulatory certifications and restricted zones.

Optical engineering requirements. Template Environment specification extends beyond object registration to **perceptual condition control**. The architecture prescribes:

- **Circular polarizing filter** on the camera lens to eliminate specular reflections at the sensor. Specular glare is polarized light; a polarizer strips it, leaving only diffuse surface appearance. Cost: $15–50.
- **Diffuse lighting** (LED ring lights, dome panels) rather than directional point sources, eliminating harsh specular highlights at the source.
- **Matte workspace surfaces** to reduce secondary reflections from the bench, bin walls, and fixtures.

This $100 environmental investment eliminates an entire class of hash instability at the physics level rather than in software. Combined with Object-Conditioned Hash Ensembling, the Template Environment provides a perceptual foundation where hash stability is a *physical guarantee*, not a statistical hope.

Economics. A mid-complexity Template Environment (200 distinct objects) can be fully digitized, semantically analyzed by a frontier VLA, and cached into the local PCO registry by a single technician in under 4 hours, with total cloud inference costs not exceeding $15 USD. The technician places objects on a $50 photography turntable, captures 36 images per object (3 elevations × 12 azimuths), and runs a single VLA inference per object for baseline SKB generation. The remaining 35 views are hashed locally on CPU and mapped to the same SKB. This represents a >99.9% reduction in time and capital expenditure compared to the 10,000+ teleoperated demonstrations required to fine-tune a standard end-to-end VLA for the same workspace.

4.8 Adaptive Hash Threshold

The hash match threshold τ is not a static tuning parameter but a **first-class architectural component**—a dynamic function of the robot's state and environmental context:

$$\tau(t) = f(\text{task_precision}(t),\ v_{\text{tcp}}(t),\ d_{\text{obstacle}}(t),\ C_{\text{env}}(t)) \tag{6}$$

where v_{tcp} is the tool center point velocity, d_{obstacle} is proximity to nearest obstacle, and C_{env} is an environmental complexity metric (e.g., visual clutter score).

Application-specific calibration:

Application	**Threshold τ**	**Rationale**
Industrial bin picking	≤10 bits (≥84% similarity)	Oil, grease, lighting variation
Warehouse logistics	≤8 bits (≥87% similarity)	Labels partially obscured
Kitchen manipulation	≤6 bits (≥90% similarity)	Variable food state
Surgical handoff	≤2 bits (≥96% similarity)	Zero margin for misidentification

The AdaptiX framework [12] validates this approach in assistive robotics, demonstrating that adaptive threshold control outperforms static thresholds across all tested scenarios. A static τ is a single point of failure: too loose invites false positives (wrong action on similar-looking object); too tight produces excessive System 2 fallback (negating the speed advantage).

4.9 Control Barrier Function Safety

Hash matching and secondary verification provide *epistemological safety*: confidence that the correct object has been identified. They do not provide *kinematic safety*: a guarantee that the commanded action will not harm people, equipment, or the workpiece. Control theory reviewers rightly demand physical guarantees.

The SKB's Safety Layer transcends semantic suggestion by explicitly parameterizing low-level **Control Barrier Functions** (CBFs). Let $x \in \mathbb{R}^n$ be the robot joint state, $u = \dot{x}$ the velocity control input, and $p(x)$ the end-effector position via forward kinematics.

Spatial keep-out zones. Given the SKB's `restricted_zones` with obstacle center p_{obs} and safety radius R_{safe}, the kinematic CBF is:

$$h_{\text{space}}(x) = \|p(x) - p_{\text{obs}}\|^2 - R_{\text{safe}}^2 \geq 0 \tag{7}$$

The low-level Real-Time Operating System (RTOS) executes the nominal System 1 trajectory u_{nom} via a 1 kHz safety-filter Quadratic Program:

$$\min_u \frac{1}{2}\|u - u_{\text{nom}}\|^2 \quad \text{s.t.} \quad 2(p(x) - p_{\text{obs}})^T J(x)\, u \geq -\gamma\, h_{\text{space}}(x) \tag{8}$$

where $J(x)$ is the analytic Jacobian and $\gamma > 0$ is a decay rate controlling barrier aggressiveness.

Crucially, while the SKB provides *static* environmental keep-out zones via its payload, the RTOS continuously ingests the raw, high-frequency depth stream to dynamically inject unmapped obstacles (e.g., human workers, dropped tools) into the CBF-QP as emergent constraints. Thus, even if System 1's semantic attention is fixated exclusively on the target ROI crop, the robot's lower-level safety layer maintains a 1 kHz reactive forcefield against dynamic intrusions into the workspace.

Pre-grasp geometric gating. To eliminate the circular dependency where a false-positive hash collision loads an unsafe CBF envelope (e.g., mistaking a fragile glass vial for a steel bolt), the system enforces a physical checksum prior to CBF parameter loading. The Kinematic ROI Isolator extracts the live bounding-box dimensions from the depth point cloud. These physical measurements are compared against the `dimensions` payload of the retrieved SKB. If the physical volume deviates beyond a configurable safety margin, execution is vetoed before the SKB's force and safety parameters are loaded. Furthermore, **proprioceptive pre-tensioning** ensures that if initial contact stiffness ($K = \Delta F/\Delta x$) measured during the first millimeter of contact does not match the expected material profile from the SKB, the impedance controller trips a hardware-level anomaly flag before absolute force limits are reached.

Force envelope. The SKB's `force_profile` (max_grip_newtons, lateral_torque_limit_nm) maps directly to hardware torque limits enforced by the robot's impedance controller. If a catastrophic false-positive hash collision causes System 1 to attempt a trajectory through a solid object, the hardware torque limit acts as a deterministic tripwire, triggering an immediate stop before damage occurs.

Velocity and torque bounds. While spatial keep-out zones require derivative-based CBFs on the system *state*, the SKB's `max_speed_mm_s` ($v_{\max}$) and torque limits map directly to convex constraints on the control *input* space $\mathcal{U}$. The QP is thus subject to the additional input bounds:

$$\|J(x)u\| \leq v_{\max}, \quad \tau_{\min} \leq M(x)\dot{u} + C(x,u)u + g(x) \leq \tau_{\max} \tag{9}$$

where $M(x)$ is the inertia matrix, $C(x,u)$ the Coriolis/centrifugal terms, and $g(x)$ the gravity vector.

The critical insight: even if perception fails completely (catastrophic hash collision), the RTOS-level safety-filter QP mathematically intercepts any unsafe command. Spatial errors are projected to the CBF boundary; kinematic errors are clamped by the convex input bounds. **Perception errors are strictly contained by deterministic physical envelopes.** This separation of concerns—epistemological safety via hashing, kinematic safety via CBFs and input bounds—enables ISO 10218-1 certification pathways that probabilistic end-to-end VLA models cannot achieve.

Impedance compliance as a mechanical fuse. For deformable objects (e.g., fluid bags, textiles) where the geometric state drifts mid-manipulation, software-level exception handling incurs unavoidable latency. PCO addresses this at the physics layer via the SKB's `compliance_stiffness` parameters. The low-level controller models the end-effector as a virtual mass-spring-damper system:

$$M_d\ddot{e} + D_d\dot{e} + K_d e = F_{\text{ext}} \tag{10}$$

where e is the trajectory error, K_d is the stiffness matrix, D_d is damping, M_d is virtual inertia, and F_{ext} is the external contact force. By setting K_d low for objects classified as `Deformable` in the Semantic Topology layer, the robotic arm acts as a mechanical shock absorber. If the object shifts or deforms mid-grasp, the physical compliance *instantaneously* ($\ll$1 ms latency) yields to the deformation, preventing damage. Simultaneously, the RTOS monitors $\|e\|$ and $\|F_{\text{ext}}\|$ at 1 kHz. If the deformation exceeds the expected compliance envelope, a proprioceptive exception (Section 4.1) is triggered within $\leq$2 ms, safely halting the arm and waking System 2 to reassess the drifted state. The physical impedance acts as a mechanical fuse, buying the necessary milliseconds for the cognitive architecture to react.

4.10 Decoupled Convergence and SKB Versioning

When a novel object is introduced to the environment, convergence occurs in two distinct phases:

Phase 1 — Visual convergence (1 trial). On first encounter, System 2's hash miss triggers VLA inference and SKB generation. Write-Back Augmentation immediately populates the registry with 30+ augmented hash keys. On the second encounter, System 1 achieves a visual hash hit with high probability. Visual recognition converges in a single System 2 cycle.

Phase 2 — Kinematic convergence (2–8 trials). The first SKB (v1.0) may contain imperfect kinematic parameters—grip force too weak, approach angle suboptimal, compliance too stiff. Physical execution reveals these errors through **proprioceptive exceptions**: the force/torque sensor detects object slip, unexpected contact, or torque limit exceedance. These proprioceptive triggers activate System 2 independently of hash state.

System 2 analyzes the tactile failure, retains the existing perceptual hash keys, and publishes an updated SKB version (v1.1) via the `amendments` field, modifying only the kinematic parameters (e.g., increasing `max_grip_newtons` from 5.0 to 15.0). This is **decoupled convergence**: visual recognition converges geometrically (1 trial via augmentation), while kinematic parameters converge physically (gradient descent in action space via impedance feedback over 2–8 trials).

4.11 Cross-Fleet Learning

When Robot A encounters a hash miss and generates a new SKB, the architecture propagates this knowledge to Robots B through Z via Zenoh, a zero-overhead pub/sub protocol designed for robotic fleet communication [26]. The propagation includes the full Write-Back Augmentation set and calibrated hash weights, so receiving robots gain both the SKB and its geometric robustness envelope.

The SKB schema includes an `amendments` field—an ordered list of annotations that accumulate over the SKB's lifetime. This enables **Operator Whisper**: human operators can annotate SKBs with domain knowledge (e.g., "Part casting supplier changed. Increased flash on edges causing occasional snags. Monitor torque spikes."). Amendments carry trust tags (`verified_institution`, `verified_human`, `inferred_model`) and visibility scopes (`fleet_wide`, `local`, `supervisor_only`).

This creates a **zero-shot fleet editability** model. If a robot consistently grips a fragile part too hard, a human operator modifies the SKB's `max_grip_newtons` field—a 5 KB JSON patch—and the entire fleet is updated via Zenoh within 100 ms. By contrast, correcting the same behavior in an end-to-end VLA requires collecting failure demonstrations, retraining the model weights, and redeploying gigabytes of parameters across the fleet.

Fleet integrity via PEST cryptographic infrastructure. To prevent malicious payload injection ("poisoned blueprints"), fleet propagation incorporates the PEST framework's cryptographic verification. Every SKB update is signed by the originating node via Public Key Infrastructure (PKI). Upon receipt, System 1 computes the `Soulmark` (SHA-256 payload hash)

and validates the signature against the Fleet Authority's certificate chain. If the signature is invalid or the Soulmark indicates tampering with CBF safety parameters (force limits, restricted zones), the SKB is quarantined and a fleet-wide alert is issued. The registry cannot be poisoned without the private cryptographic keys of the Fleet Authority—a property inherited directly from the PEST protocol's four-state provenance machine (Pristine / Transformed / Tampered / Unresolvable) [14].

4.12 Extension to Dynamic Objects

PCO is optimized for **quasi-static Template Environments** where objects are stationary or slow-moving. In fully dynamic scenarios—objects tumbling down a chute, being handed over by a human, or rolling after a failed grasp—the 2D projection changes every frame, and no pre-computed hash template will match reliably. For these scenarios, the architecture degrades gracefully to continuous System 2 operation.

However, we identify a promising extension: **PCA canonicalization** of the RGB-D point cloud. Using the isolated 3D point cloud from the Kinematic ROI Isolator, a microsecond-scale Principal Component Analysis identifies the object's three primary axes of inertia. To resolve the inherent sign ambiguity of PCA eigenvectors (the 180° flip problem), we enforce axis directionality using the third-order central moments (skewness) of the point cloud distribution along each principal axis. The point cloud is then rotated so its primary axis aligns with the camera's Z-axis, creating a normalized orthogonal projection. The RGB texture is projected onto this canonical plane and hashed, yielding a representation that is approximately $SO(3)$ rotation invariant.

This approach requires careful validation—PCA on partial point clouds can be unstable for symmetric objects (cylinders, spheres)—and we reserve it for Phase 2 of the experimental program. The quasi-static case is sufficient for the vast majority of industrial manipulation tasks.

5. Experimental Design

We specify a four-phase validation protocol, implemented in the open-source `codex_lab_kit` [34] and distributable to partner laboratories. The robotic integration layer is provided by the `gcp-robotics` SDK [35]. The protocol uses YCB benchmark objects [21] for standardization and reproducibility.

5.1 Phase A: Hash Robustness (Camera Only)

Objective: Validate that composite perceptual hashing discriminates objects reliably under realistic visual variation, including with the Kinematic ROI Isolator pipeline.

Setup: 20 YCB objects × 12 viewpoints (30° turntable increments) × 3 lighting conditions (ambient, overhead bright, directional side) = **720 images**. No robot arm required. Images captured both as full-frame and as depth-gated ROI crops to validate the isolation pipeline.

Metrics:

- Intra-class Hamming distance (mean, std): same object across viewpoints/lighting
- Inter-class Hamming distance (mean, std): different objects
- ROC curves at thresholds 0–19 bits (per hash type and weighted composite)
- Recall and false positive rate at threshold 5
- Per-hash variance analysis across augmentation set

Acceptance criteria:

- Intra-class Hamming mean < 6 bits (composite weighted distance)
- Inter-class Hamming mean > 15 bits
- Recall at threshold $5 > 0.90$
- False Acceptance Rate $< 0.1\%$

Estimated duration: $\sim$50 minutes.

5.2 Phase B: Hash-to-Grasp Correlation (Full Hardware)

Objective: Establish that hash-indexed SKBs transfer reliably to physical manipulation.

Setup: 10 objects, 10 grasps per object, recording binary success, actual force profile, and execution time. Hash distance at lookup time is correlated with grasp success.

Metrics:

- Grasp success rate by Hamming band (0–2, 3–5, 6–8, 9+)
- Success by match type (EXACT vs. FUZZY)
- Mean force deviation from SKB-prescribed profile (N)
- Post-action hash shift (validates execution success)
- False acceptance events (wrong SKB executed)

Acceptance criteria:

- SKB-indexed grasp success within 2% of VLA baseline
- False Acceptance Rate $< 0.1\%$

Estimated duration: $\sim$200 minutes.

5.3 Phase C: Loop Closure Convergence (Full Hardware)

Objective: Measure how quickly the system learns novel objects through the System 1/System 2 feedback loop, tracking visual and kinematic convergence separately.

Setup: 5 novel objects (not in initial registry), 50 consecutive trials each. System 2 generates SKBs on hash miss; Write-Back Augmentation promotes them to System 1. Proprioceptive exceptions trigger kinematic refinement.

Metrics:

- Visual convergence: trials to first System 1 hash hit
- Kinematic convergence: trials to 90% grasp success
- Total VLM calls vs. hash lookups (convergence curve)
- SKB version count (amendments for kinematic refinement)

Acceptance criteria:

- Visual convergence in $\leq$2 trials (hash hit after first Write-Back)
- Kinematic convergence: median $\leq$8 trials to 90% grasp success
- Final success rate > 0.90

Estimated duration: ∼250 minutes.

5.4 Phase D: Latency Profile (Full Hardware)

Objective: Measure end-to-end timing breakdown and compare against neural baselines.

Setup: 5 objects, 50 trials each (plus 10 warmup iterations). Timing components measured independently.

Timing breakdown:

- Kinematic ROI Isolation (ms)
- Hash computation (ms)
- Registry lookup (ms)
- Secondary verification (ms)
- ROS2 transport (ms)
- CBF-QP safety filter (ms)
- Total perception-to-action (ms)

Baseline comparison: CLIP feature matching pipeline (encode → cosine similarity → action retrieval).

Acceptance criteria:

- Hash lookup $p_{99} < 1.0\,\text{ms}$
- Total fast-path $p_{99} < 10.0\,\text{ms}$

Estimated duration: ∼30 minutes.

5.5 Adversarial Stress Tests

To rigorously ensure the architecture does not succumb to the failure modes of classical template matching, Phases B and C incorporate explicit stress tests:

- **Micro-Increment Rotational Test:** 0.5° turntable increments across ±15° to verify that continuous kinematic anchoring (Section 4.3) prevents step-function grasping errors at template boundaries.
- **Specular Stress Test:** Dynamic directional lighting on machined metal cylinders to validate zHash invariance and Object-Conditioned Hash Ensembling under worst-case specular gradient inversion.
- **Poisoned Hash Protocol:** Intentionally injecting false-positive hash matches (mapping an object to the wrong SKB) to validate that geometric gating and CBF interception prevent unsafe execution.
- **Deformable State Drift:** Grasping compliant objects (silicone bags, textiles) to validate that impedance compliance (Section 4.9) absorbs mid-grasp deformation before proprioceptive exception triggers.

5.6 Hardware Requirements

The protocol is designed to be accessible to standard university robotics labs:

Component	Minimum	Ideal
Robot arm	6-DOF (UR5e, Franka Panda)	7-DOF with impedance control
Gripper	Parallel jaw with position feedback	Robotiq 2F-85 with F/T sensing
Camera	RGB-D (RealSense D435)	+ circular polarizing filter
Compute	ARM CPU (hash) + GPU (VLA)	+ FPGA for μs hashing
Lighting	Controlled overhead	Diffuse LED dome/ring
Environment	Lab bench	Industrial test cell
Personnel	1 graduate student	PI + 2 students

6. Expected Results and Falsification Criteria

6.1 Expected Results

Based on the SKB worked example (fast-path latency 0.8 ms, slow-path latency 1835 ms) and the architectural analysis:

- **Latency reduction**: <6 ms fast-path (ROI isolation + hash + lookup + verification) vs. 500–2000 ms VLA inference = 97–99.7% reduction
- **System 1 hit rate**: >95% in a well-populated Template Environment after convergence
- **Grasp success**: Within 2% of VLA baseline for exact hash matches
- **Visual convergence**: 1 trial (Write-Back Augmentation populates 30+ hash keys)
- **Kinematic convergence**: $\leq$8 trials via proprioceptive feedback and SKB versioning
- **Fleet learning**: New SKBs available to all fleet robots within 100 ms of promotion (Zenoh transport)
- **Edge deployment**: System 1 runs on ARM CPU at <2 W, no GPU required

6.2 Falsification Criteria

This is a hypothesis paper. We specify the conditions under which each hypothesis is **rejected**:

Hypo.	Falsified If	Implication
H1	Total fast-path $p_{99} > 10$ ms, OR SKB grasp success degrades >2% vs. VLA baseline	Insufficient speed or reliability advantage over Generation 2 VLAs
H2	Visual convergence requires >2 trials, OR kinematic convergence requires >8 trials to reach 90%	Write-Back Augmentation or proprioceptive refinement loop fails
H3	False Acceptance Rate > 0.1% in Phase B, resulting in execution of incorrect action primitives	Hash-indexed control introduces unacceptable safety risk despite CBF safeguards
All	Intra-class weighted Hamming distances overlap inter-class distances in Phase A	Hashes are fundamentally not discriminative for the object set

We note that while a raw visual False Acceptance Rate of 0.1% equates to 1,000 Defects Per Million Opportunities (DPMO)—above typical Six Sigma thresholds—the *system-level* failure rate is multiplicatively lower due to layered defense. A catastrophic execution error requires the intersection of a visual false positive ($P \approx 10^{-3}$), a geometric gating failure where the point-cloud

dimensions pass the volume check ($P \approx 10^{-3}$), and a proprioceptive pre-tensioning failure where contact stiffness matches the wrong material ($P \approx 10^{-3}$). The joint probability ($\sim 10^{-9}$) places the architecture within Safety Integrity Level 2 (SIL 2) thresholds for industrial deployment. The 0.1% criterion above tests the *raw perceptual layer* in isolation; the full safety stack provides additional orders of magnitude of protection.

We commit to publishing negative results. If the experimental protocol produces falsification evidence, the results paper will report this transparently. The community benefits equally from knowing what does *not* work.

7. Partnership Model

7.1 What Metavolve Brings

Asset	Description	Status
PEST Framework	Hash-to-metadata architecture validated on 10,097 artifacts	Production
Patent Portfolio	U.S. Prov. App. No. 63/983,304 (PEST/PCO core) + U.S. Prov. App. No. 63/984,299 (advanced PCO subsystems); 21+ claims covering hash ensembling, kinematic isolation, CBF safety, fleet integrity	Filed
CompositeHasher	Multi-hash engine (dHash, pHash, pHash-256, color hash)	Implemented
SKB Schema	Four-layer Pydantic v2 model (742 lines) with CBF parameterization	Implemented
`codex_lab_kit` [34]	Standardized validation toolkit with CalibrationWizard, ExperimentProtocol, DataCollector, ResultsAnalyzer	Implemented
`gcp-robotics` SDK [35]	Robotic integration layer for PCO pipeline	Implemented
Published Research	3-paper trilogy: +160% semantic coverage from dense metadata	Published
Template Registrar	Automated capture-hash-register workflow ($15 / 4 hours per 200-object workspace)	Implemented

7.2 What Partners Bring

- Robotic hardware (6-DOF+ arm, parallel-jaw gripper with force sensing, RGB-D camera)
- Real-world test environments (assembly cell, kitchen, lab bench)
- Domain expertise in specific application verticals
- Graduate student or postdoctoral effort for data collection and analysis

7.3 IP Structure

- **Metavolve retains**: PEST framework IP, patent rights (Claims 15–24), hash engine, core protocol architecture
- **Partner retains**: Domain-specific SKBs generated during validation, Template Environment configurations, novel grasp primitives

- **Joint**: All publications are co-authored. Experimental data is jointly owned.
- **Open science**: The `codex_lab_kit` remains open source. Raw experimental data and analysis scripts are released with publications.

8. Related Work

Table 2: Comparison of approaches to the Inference-Control Discrepancy.

System	Gen.	Speed	Caching	GPU?	Determ.	Editable
RT-2	High	1–3 Hz	None	Cloud	No	No
π_0-FAST	High	50–200 Hz	None	Desktop	No	No
HyperVLA	High	>100 Hz	Implicit	Desktop	No	No
MeCo	Med	Variable	Plans	Cloud	Partial	No
PCO	**High**	**>1 kHz**	**SKB**	**No**	**Yes**	**Yes**

PCO is **complementary** to VLA models, not a replacement. System 2 *is* a VLA model. The innovation is knowing *when to skip it*—and providing a principled mechanism (hash-indexed retrieval) for doing so. Three dimensions differentiate PCO from even the fastest Generation 2 architectures:

The thermodynamic wall. Achieving 4 ms on HyperVLA requires an NVIDIA RTX 4090 or Hopper GPU at the edge—300+ Watts, active cooling, thousands of dollars. PCO's $\mathcal{O}(1)$ hash lookup runs on a standard embedded ARM CPU or a $10 FPGA at <2 Watts. For untethered mobile manipulators where battery life is a binding constraint, PCO is not merely faster—it is the only viable architecture.

ISO-compliant determinism. Neural networks are probabilistic. Even at 4 ms, a VLA can hallucinate a destructive trajectory on its 10,000th cycle. PCO's Fast Path is 100% deterministic: a hash match guarantees bit-for-bit trajectory replay. An ISO 10218-1 safety certification can be obtained for a deterministic hash registry backed by CBF safety filters. Certifying an unconstrained probabilistic model for a factory floor remains an open problem.

Zero-shot fleet editability. If π_0-FAST consistently grips a fragile part too hard, the correction requires collecting failure data, retraining model weights, and redeploying gigabytes of parameters across the fleet. With PCO, a human line manager modifies the SKB's `max_grip_newtons` field—a 5 KB JSON patch—and the entire fleet is updated via Zenoh within 100 ms.

The closest related work is MeCo [13], which applies plan memoization to multi-robot LLM collaboration. Our contribution extends this principle from task plans to manipulation primitives, replaces semantic similarity with perceptual hashing for speed, and introduces the four-layer SKB schema for encoding physical manipulation knowledge.

Recent demonstrations of VLAs performing highly dynamic manipulation (e.g., catching thrown objects, dynamic handover) implicitly validate the PCO hypothesis. To achieve these feats, researchers bypass continuous VLA inference during the execution phase, relying instead on high-frequency, non-neural kinematic trackers or distilled state-action policies once the semantic target is identified [32]. PCO formalizes this ad-hoc workaround into a principled, mathematically rigorous cognitive architecture, replacing opaque, compute-heavy trackers with a deterministic, $\mathcal{O}(1)$ perceptual hash registry.

Grasp databases (DexNet [22], Columbia Grasp Dataset) store pre-computed grasps but lack perceptual indexing and runtime retrieval. Object recognition systems (FoundationPose [23], SAM [24]) provide detection but do not bridge to action schemas. Perceptual hashing has been

applied to visual tracking [25] but not, to our knowledge, as a primary indexing mechanism for robotic manipulation plans.

9. Conclusion and Call for Collaboration

We have presented **Perceptual Compute Offloading**: a hypothesis for resolving the Inference-Control Discrepancy in VLA-driven robotics through a dual-process architecture that uses hash-indexed Spatial Kinematic Blueprints to achieve sub-millisecond perception-to-action latency for known environmental states.

The architecture is grounded in four pillars: (1) the empirically validated PEST framework, which demonstrated that dense structured metadata produces measurably superior model behavior; (2) convergent evidence from independent research groups (AdaDrive, AdaptiX, MeCo, HyperVLA) arriving at the same fast/slow architectural bifurcation; (3) a formal experimental protocol with explicit falsification criteria; and (4) mathematical safety guarantees via Control Barrier Functions that contain perception errors within deterministic physical envelopes.

We explicitly bound the scope of this architecture. While end-to-end VLAs often target the high-entropy domain of domestic "home robotics"—where zero-shot generalization supersedes speed—PCO is architected for the high-frequency, zero-tolerance demands of structured and semi-structured industrial Template Environments (manufacturing, biopharma, logistics, surgical). Extension of hash-indexed reflexes to fully unstructured, highly chaotic consumer environments remains a subject for future research. By accepting this domain constraint, PCO secures the deterministic reliability required for enterprise deployment.

PCO offers three properties that no Generation 2 VLA architecture can match: **zero-GPU edge deployment** (<2 W on ARM), **ISO-certifiable determinism** (bit-for-bit trajectory replay), and **zero-shot fleet editability** (5 KB JSON patch updates the entire fleet in 100 ms). A Template Environment can be fully populated for $15 in 4 hours—a >99.9% reduction versus the teleoperated demonstrations required for end-to-end VLA training.

What remains is **hardware validation**. The CompositeHasher works. The SKB schema is formal. The CBF safety formulation is standard. The experimental protocol is codified. What we need is a robotic arm, a camera, and a partner willing to run the experiment.

This paper is an invitation. The `codex_lab_kit` [34] is available to any laboratory with standard robotic manipulation hardware. The first partner labs to validate or falsify these hypotheses will be co-authors on the results publication. If the architecture works as hypothesized, it represents a path to deploying VLA-level intelligence at industrial control frequencies—bridging the gap between what foundation models know and how fast robots need to act.

The PCO architecture and its subsystems are protected under pending U.S. provisional patent applications assigned to Metavolve Labs, Inc. Collaboration inquiries and licensing discussions are welcome.

Contact: `research@iaeternum.ai`

References

[1] Texas Instruments, "An engineer's guide to industrial robot designs," 2023. Available: https://www.ti.com/lit/eb/ssiy006/ssiy006.pdf

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

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

[4] Octo Model Team, "Octo: An Open-Source Generalist Robot Policy," *arXiv:2405.12213*, 2024.

[5] Xiaomi Robotics, "Xiaomi-Robotics-0: An Open-Sourced Vision-Language-Action Model with Real-Time Execution," *arXiv:2602.12684*, 2026.

[6] Physical Intelligence, "VLAs that Train Fast, Run Fast, and Generalize Better," 2025. Available: https://www.pi.website/research/knowledge_insulation

[7] "HyperVLA: Efficient Inference in Vision-Language-Action Models via Hypernetworks," *arXiv:2510.04898*, 2025.

[8] "Vision-Language-Action (VLA) Models: Concepts, Progress, Applications and Challenges," *arXiv:2505.04769*, 2025.

[9] D. Kahneman, *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

[10] K. E. Stanovich and R. F. West, "Individual differences in reasoning: Implications for the rationality debate?" *Behavioral and Brain Sciences*, vol. 23, no. 5, pp. 645–665, 2000.

[11] Zhang et al., "AdaDrive: Self-Adaptive Slow-Fast System for Language-Grounded Autonomous Driving," *ICCV*, 2025.

[12] "Embedded Intelligence for Shared Control: Threshold Tuning and AI Runtime Switching in Assistive Human-Robot Interaction," 2025.

[13] "MeCo: Enhancing LLM-Empowered Multi-Robot Collaboration via Similar Task Memoization," 2025.

[14] T. MacPherson, "The Entropy of Recursion: A Strategic Framework for Provenance-Verified Data and the Preservation of Signal Integrity in Generative Systems," Metavolve Labs, 2025. DOI: 10.5281/zenodo.18436975.

[15] T. MacPherson, "The Density Imperative: How Semantic Curation Depth Determines Vision-Language Model Capability," Metavolve Labs, 2026. DOI: 10.5281/zenodo.18667735.

[16] T. MacPherson, "Cognitive Nutrition: Architecture of a Provenance-Embedded Data Refinery," Metavolve Labs, 2026. DOI: 10.5281/zenodo.18667742.

[17] M. Norouzi et al., "Fast Exact Search in Hamming Space with Multi-Index Hashing," *IEEE TPAMI*, vol. 36, no. 6, pp. 1107–1119, 2014.

[18] "Evaluating robustness of perceptual image hashing algorithms," 2017. Available: https://www.researchgate.net/publication/318696121

[19] "It's Not What It Looks Like: Manipulating Perceptual Hashing based Applications," University of Illinois.

[20] "A fast monocular 6D pose estimation method for textureless objects using template matching," *PMC*, 2025.

[21] B. Calli et al., "The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research," *IEEE ICAR*, 2015.

[22] J. Mahler et al., "Dex-Net 2.0: Deep Learning to Plan Robust Grasps," *RSS*, 2017.

[23] B. Wen et al., "FoundationPose: Unified 6D Pose Estimation and Tracking of Novel Objects," *CVPR*, 2024.

[24] A. Kirillov et al., "Segment Anything," *ICCV*, 2023.

[25] "Real-time Visual Tracking Based on Improved Perceptual Hashing," University of Portsmouth.

[26] R. Hoque, "Scalable Lifelong Imitation Learning for Robot Fleets," UC Berkeley, 2024.

[27] "Shaping Cognitive Control for HRI through the Dual-Process Theory," *CEUR-WS*, 2023.

[28] "Adversarial collision attacks on image hashing functions," *AML@CVPR Workshop*, 2021.

[29] "HOMI: Ultra-Fast EdgeAI platform for Event Cameras," 2025.

[30] "Experiences from Benchmarking Vision–Language–Action Models for Robotic Manipulation," *arXiv:2511.11298*, 2025.

[31] A. D. Ames et al., "Control Barrier Functions: Theory and Applications," *European Control Conference*, 2019.

[32] "A Vision-Language-Action Model for Dynamic Object Manipulation," 2026. Available: https://www.youtube.com/watch?v=1h922LfBJnQ

[33] T. MacPherson, "Alexandria Aeternum Dataset," 2025. DOI: 10.5281/zenodo.18359131. Available: https://huggingface.co/datasets/codex-curator/alexandria-aeternum

[34] T. MacPherson, "codex-lab-kit v1.0.1: Standardized Validation Toolkit for Perceptual Compute Offloading," 2026. DOI: 10.5281/zenodo.18668110. Available: https://github.com/codex-curator/codex-lab-kit

[35] T. MacPherson, "gcp-robotics SDK: Golden Codex Pipeline Robotics Integration," 2026. DOI: 10.5281/zenodo.18668113. Available: https://github.com/codex-curator/gcp-robotics

Patent Disclosure

Elements of the architecture described in this paper are the subject of two U.S. Provisional Patent Applications: (1) No. 63/983,304, "Systems and Methods for Provenance-Embedded Semantic Transport, Multi-Agent Data Enrichment, and Perceptual Compute Offloading in Autonomous Systems," Claims 15–24; and (2) No. 63/984,299, "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," Claims 1–21. Both applications are assigned to Metavolve Labs, Inc. The experimental protocol and `codex_lab_kit` are released under the Apache 2.0 license for research and commercial use.

Built by Metavolve Labs (https://iaeternum.ai) — "The factory is open."