

# Snath Robotics: Multi-Stream Divergence Routing for Humanoid Robotics

Aadithya Vishnu Sajeev<sup>†</sup>

*Snath AI — Open Source Research Initiative*

[axdithya@snath.ai](mailto:axdithya@snath.ai) ORCID: 0009-0009-3916-0988

<https://github.com/snath-ai/snath-robotics>

Zenodo. [doi:10.5281/zenodo.20517446](https://doi.org/10.5281/zenodo.20517446). June 2026.

Code: Apache 2.0 | Paper: CC BY 4.0

<sup>†</sup> Correspondence: [axdithya@snath.ai](mailto:axdithya@snath.ai)

## Abstract

We formalise the application of the V1–V6 cognitive routing contract [1, 2] to humanoid robotics, constituting the fourth domain instantiation of the Lár-JEPA architecture [3]. The architecture maintains two structurally independent latent streams—visual appearance ( $z_{\text{vision}} \in \mathbb{R}^D$ ) and proprioceptive physics ( $z_{\text{proprio}} \in \mathbb{R}^D$ )—that are *never fused*. A mathematically frozen divergence router measures their total-variation distance  $D = \|\text{softmax}(z_A) - \text{softmax}(z_B)\|_1 / \sqrt{G}$  and routes to one of four decisions: COMMIT, REPLAN, IMPASSE, or DEFER. An overnight DMN consolidation cycle trains signed LoRA [9] adapters from historical sensor-disagreement events and distributes them to the fleet, implementing a swarm learning mechanism in which a single failure event can improve the behaviour of every deployed unit.

We make three contributions: (i) a formal mapping from the M1–M3 encoder invariants to sensor modalities; (ii) robotics-specific temporal decay constants for three failure classes (`environmental_transient`,  $\lambda=0.50$ ; `sensor_drift`,  $\lambda=0.20$ ; `hardware_structural`,  $\lambda=0.02$ ); and (iii) a precise statement of the boundary between *routing safety*—what the architecture guarantees by construction—and *actuation safety*—what requires integration with a physics or model-predictive control layer.

This paper does not report empirical results; it establishes the theoretical mapping and constitutes prior art for the robotics domain application of the architecture. The reference implementation is available at <https://github.com/snath-ai/snath-robotics> under Apache 2.0.

**Keywords:** Humanoid Robotics, Cognitive Architecture, Divergence Routing, Multi-Stream Routing, Joint-Embedding Predictive Architecture, Lár-JEPA, Domain Isomorphism, Temporal Decay, LoRA Adapter, Swarm Learning, System 1/System 2, Safety-Learning Equivalence.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	The Lár-JEPA Architecture . . . . .	3
2.2	The V1–V6 Routing Contract . . . . .	4
<b>3</b>	<b>Sensor-to-Stream Mapping (M1–M3)</b>	<b>4</b>
3.1	Structural Independence . . . . .	4
3.2	The Canonical Disagreement Scenarios . . . . .	4
<b>4</b>	<b>Temporal Decay and the Identification/Correction Asymmetry</b>	<b>5</b>
4.1	The Temporal Trust Gate . . . . .	5
4.2	Identification/Correction Trust Asymmetry . . . . .	5
<b>5</b>	<b>The Physics Layer Distinction</b>	<b>6</b>
<b>6</b>	<b>The DMN Overnight Consolidation Cycle</b>	<b>6</b>
6.1	The $D_{\text{hard}}$ Curriculum . . . . .	6
6.2	Consolidation Cycle . . . . .	6
6.3	Fleet Distribution . . . . .	7
6.4	SIGREG Integration . . . . .	7
<b>7</b>	<b>Domain Isomorphism</b>	<b>7</b>
<b>8</b>	<b>Limitations and Future Work</b>	<b>7</b>
<b>9</b>	<b>Conclusion</b>	<b>8</b>

## 1 Introduction

Current humanoid robots sit at two architectural extremes, each with a characteristic failure mode.

**End-to-end deep learning** (e.g., vision-language-action models [11]) fuses all sensor modalities into a single learned function. When any input stream is corrupted—cameras blinded by sun glare, friction lost to ice—the corruption propagates through every downstream decision without structural containment. The model that confidently predicted a safe floor continues to command a forward step because it has no mechanism to recognise that its own inputs contradict each other.

**Classical control** systems (e.g., model-predictive control with rigid dynamics models) are deterministically safe but learn only through manual re-engineering of the model and constraint set. A robot encountering a novel failure mode cannot update its behaviour without an engineer in the loop.

We argue that neither failure is intrinsic to the problem. Both follow from the same architectural choice: either fuse all streams into a single function (and lose the disagreement signal) or operate with a fixed model (and lose the learning signal). The Lár-JEPA routing architecture [1, 2] provides a third path: structurally independent streams with a frozen routing contract that neither fuses nor discards disagreement, and an overnight consolidation cycle that learns from disagreement events without modifying the frozen router.

### Contributions.

1. A formal mapping from the M1–M3 encoder independence invariants (Section 3) to humanoid sensor modalities: visual appearance (Stream A) and proprioceptive physics (Stream B).
2. A domain instantiation of the V1–V6 divergence routing contract (Section 2.2) with robotics-specific failure classes and temporal decay constants (Section 4).
3. A precise statement of the boundary between routing safety and actuation safety (Section 5), which bounds what the architecture can and cannot guarantee.
4. A reference implementation at <https://github.com/snath-ai/snath-robotics> with 7/7 regression tests passing.

**Scope.** This is a position paper. It establishes the theoretical mapping and prior art. Empirical validation is deferred to AIA Experiment 3 [3] (SIGReg routing amplification) and a forthcoming hardware evaluation on a physical platform.

## 2 Background

### 2.1 The Lár-JEPA Architecture

The Lár-JEPA framework [1] defines ten Abstract Base Classes (ABCs) and 33 named invariants governing a cognitive execution spine. The ten ABCs cover world-model prediction, cross-modal adaptation (CB1–CB2), latent fault localisation (I1–I6), entropy-gated routing, pluggable attention (A1–A6), counterfactual perturbation (P1–P6), score-then-route decision (R1–R4), encoder projection (M1–M3), and content-blind divergence routing (V1–V6).

The architecture has been instantiated in three prior domains:

- **Snath Locus** (CRISPR drug screening): DNA structural geometry (Stream A) vs. patient RNA disease state (Stream B).

- **Snath Basis** (quantitative finance): fundamental analysis (Stream A) vs. market signals (Stream B).
- **Snath Aviation** (aviation sensor routing): radar altimeter (Stream A) vs. pitot tube (Stream B).

In all three cases, the routing code is byte-for-byte identical. Only the encoder constructors, failure-class labels, and temporal decay constants differ. This paper adds Snath Robotics as the fourth instantiation.

## 2.2 The V1–V6 Routing Contract

`AbstractDivergenceRouter` [2] specifies six routing invariants:

- V1** Both streams present at every routing call.
- V2** Divergence computed from normalised probability vectors only.
- V3** Routing decision is a pure function of scalar  $(D, c_A, c_B)$ .
- V4** *Content-blind*: the router never branches on the raw values of  $z_A$  or  $z_B$ —only on scalars derived from them.
- V5** `STRUCTURAL_IMPASSE` is always reachable regardless of stream content.
- V6** `COMMIT_TRAJECTORY` is returned only when  $D < \tau_{\text{low}}$  and both  $c_A, c_B \geq \tau_{\text{low}}$ .

The divergence metric is total-variation distance normalised by  $\sqrt{G}$ :

$$D = \frac{\|\text{softmax}(z_A) - \text{softmax}(z_B)\|_1}{\sqrt{G}} \quad (1)$$

where  $G$  is the shared embedding dimension. Division by  $\sqrt{G}$  ensures thresholds  $\tau_{\text{high}}, \tau_{\text{low}}$  are comparable across domains with different embedding sizes [2].

## 3 Sensor-to-Stream Mapping (M1–M3)

### 3.1 Structural Independence

The M1–M3 encoder invariants require that neither stream is aware of the other’s output prior to the routing call [1]. For humanoid robotics:

**Stream A — Visual appearance** ( $z_{\text{vision}} \in \mathbb{R}^D$ ). Camera frames processed by a visual backbone (CLIP [6], DINOv2 [7], or a robotics-specific model such as R3M [8]). This stream answers: *what does the scene look like?*

**Stream B — Proprioceptive physics** ( $z_{\text{proprio}} \in \mathbb{R}^D$ ). IMU (accelerometer, gyroscope), joint torque sensors, and fingertip pressure sensors, embedded jointly into the shared latent space. This stream answers: *what does the body actually experience?*

M1 (stream independence) is satisfied structurally: the two encoders share no parameters and read no activations from each other. M2 (shared embedding space) is satisfied by projecting both streams into  $\mathbb{R}^{768}$  via separate projection heads. M3 (no cross-stream gradient during inference) is satisfied by freezing the projection heads at inference time and training them independently during the overnight consolidation cycle.

### 3.2 The Canonical Disagreement Scenarios

**Example 1** (Ice slip). *The visual stream encodes a flat, uniformly textured floor surface. The proprioceptive stream encodes near-zero friction coefficient and lateral acceleration onset. The two probability vectors  $p_A = \text{softmax}(z_{\text{vision}})$  and  $p_B = \text{softmax}(z_{\text{proprio}})$  are substantially different:*

$D \gg \tau_{\text{high}}$ .  $V_4$  (content-blindness) ensures the router does not attempt to parse why the streams disagree—it observes only that they do, and by  $V_5$  returns `STRUCTURAL_IMPASSE`. The robot drops to a pre-programmed brace position.

**Example 2** (Motor degradation). The visual stream encodes a normal walking scene. The proprioceptive stream encodes an asymmetric joint torque response (left knee reporting  $\sim 50\%$  of commanded torque, indicating motor wear).  $\tau_{\text{low}} \leq D < \tau_{\text{high}}$ ; the router returns `TRIGGER_REPLAN`. The adapter router attempts to load a signed LoRA delta that compensates the proprioceptive encoder’s projection for this joint failure signature.

These two scenarios—irreconcilable stream contradiction and recoverable stream mismatch—are structurally identical to the scenarios handled in Snath Aviation (pitot freeze and GPS spoof) and Snath Basis (momentum vs. fundamental divergence). Only the sensor labels and physics differ.

## 4 Temporal Decay and the Identification/Correction Asymmetry

### 4.1 The Temporal Trust Gate

The temporal trust gate is  $W = \exp(-\lambda \cdot \Delta t)$ , where  $\Delta t$  is fractional years since the adapter was trained and  $\lambda$  is a failure-class constant. Adapters with  $W < W_{\text{min}} = 0.40$  are refused before System 2 injection. Table 1 gives the robotics-specific constants.

Failure class	$\lambda$	Trust half-life
<code>environmental_transient</code> (ice, glare, wet floor)	0.50	1.4 yr
<code>sensor_drift</code> (gradual calibration error)	0.20	3.5 yr
<code>hardware_structural</code> (motor wear, joint degradation)	0.02	34.7 yr

Table 1: Temporal decay constants for Snath Robotics. The formula  $W = \exp(-\lambda \cdot \Delta t)$  and the  $W_{\text{min}} = 0.40$  floor are identical to Snath Aviation [4] and Snath Basis [5]; only the failure-class labels differ.

Environmental transients age quickly because sensor conditions change with season, geography, and floor material. Hardware structural failures age slowly because motor wear is a monotonic physical process and learned corrections remain valid for the lifetime of the component.

### 4.2 Identification/Correction Trust Asymmetry

**Definition 1** (Identification/Correction Trust Asymmetry). System 1 centroid matching (identification) is trust-invariant: *the geometric fingerprint of a failure class is durable across years*. System 2 LoRA injection (correction) is perishable: *the learned delta is gated by  $W \geq W_{\text{min}}$  before injection*.

This asymmetry is formalised in [3], §3.4 Remark (Temporal Decay and Synaptic Depression), and extended to the robotics domain here.

The practical consequence: when System 2 is refused due to a stale adapter, System 1 still identifies the failure class and returns `COMMIT_TRAJECTORY` with the correct lean decision. The audit note records both the identification event and the stale-adapter refusal. The intended degradation path is: *identify correctly, correct conservatively*.

**Remark 1** (Synaptic depression analogy). The temporal trust gate  $W = \exp(-\lambda \cdot \Delta t)$  is structurally identical to short-term synaptic depression in biological neural circuits [10], where synaptic efficacy decays exponentially after a period of inactivity and recovers through consolidation. The identification/correction asymmetry mirrors the distinction between semantic memory

(durable recognition of a failure type) and episodic memory (perishable recollection of the specific correction that worked last time).

## 5 The Physics Layer Distinction

The following remark precisely states the boundary of what the routing architecture guarantees, correcting a common conflation between routing safety and actuation safety.

**Remark 2** (Scope of routing guarantees). *The V1–V6 invariants are routing safety guarantees, not actuation safety guarantees.*

*The router guarantees, by construction: (i) when  $D \geq \tau_{\text{high}}$  the decision is always STRUCTURAL\_IMPASSE (V5); (ii) the decision is computed without reading stream content (V4 content-blindness); and (iii) COMMIT\_TRAJECTORY is never returned when both confidence values are below  $\tau_{\text{low}}$  (V6).*

*The router does not guarantee that the actuator commands resulting from a routing decision satisfy force, torque, or collision bounds. Preventing a robot from exerting excessive grip force, for example, requires either mechanical force limits, a real-time physics model, or model-predictive control operating on the actuator signal downstream of the routing decision.*

*The two layers are complementary. The routing layer prevents the physics layer from receiving contradictory or corrupted sensor inputs. The physics layer prevents the routing layer’s decisions from translating into unsafe motor commands. Neither replaces the other.*

This distinction is important for correctly scoping any claim about the architecture’s safety properties. The V1–V6 contract provides mathematical guarantees about the *decision function under sensor disagreement*. It does not provide mathematical guarantees about physical consequences of those decisions without a validated physics layer.

## 6 The DMN Overnight Consolidation Cycle

### 6.1 The $\mathcal{D}_{\text{hard}}$ Curriculum

Every TRIGGER\_REPLAN and STRUCTURAL\_IMPASSE event is HMAC-SHA256 signed and written to the  $\mathcal{D}_{\text{hard}}$  queue with the full latent-space snapshot at divergence time. The  $\mathcal{D}_{\text{hard}}$  curriculum from the DAS paper [2] specialises to:

$$\mathcal{D}_{\text{hard}} = \{i : D_i \geq \delta \text{ and } r_i = \text{TRIGGER\_REPLAN}\} \quad (2)$$

When a human operator or ground-truth sensor log labels which stream was correct (i.e., sets the **winner** field), the event becomes training data for the next consolidation cycle.

### 6.2 Consolidation Cycle

The overnight `RoboticsDMN consolidate()` procedure:

1. Reads all verified, resolved events from the  $\mathcal{D}_{\text{hard}}$  queue.
2. Clusters events by failure class.
3. For each cluster with  $n \geq 4$  resolved events:
  - (a) Computes the mean  $\Delta$ -vector as a System 1 JSON centroid sidecar.
  - (b) Trains a Rank-1 LoRA [9] ( $A, B$ ) delta by minimising L1 alignment loss between the faulty stream’s adapted embedding and the winner stream’s embedding, optionally augmented with the SIGREG isotropy penalty (see Section 6.4).
  - (c) HMAC-signs the .pt payload and saves to `models/adapters/`.
4. The adapter router picks up new files on the next `refresh()` call.

### 6.3 Fleet Distribution

Adapter files are self-contained, self-verifying artefacts: each carries the HMAC signature, `target_encoder` type-safety field (preventing a “pitot” adapter being injected into a radar encoder), `created_at` timestamp, and `failure_class` for temporal decay routing. A deployment system can distribute new adapters to the entire fleet by copying the `models/adapters/` directory; each robot verifies the HMAC before applying any delta. No central coordination is required beyond file distribution.

### 6.4 SIGReg Integration

The SIGREG covariance penalty from [3] is wired into the consolidation loop:

$$\mathcal{L}_{\text{SIGREG}} = \mathcal{L}_{\text{task}} + \frac{\lambda_{\text{iso}}}{D} \sum_{i \neq j} [\text{Cov}(Z)]_{ij}^2 \quad (3)$$

with  $\lambda_{\text{iso}} = 0.0$  (inert) by default. When the encoder’s latent space is anisotropic, the divergence score  $D$  (Equation 1) is dominated by a few high-variance dimensions, degrading routing reliability. SIGREG forces all  $D$  dimensions to carry comparable variance, making  $D$  a more informative routing signal. AIA Experiment 3 [3] targets  $\rho = \text{AUROC}_{\text{SIGREG}} / \text{AUROC}_{\text{base}} > 1.15$ .

## 7 Domain Isomorphism

Table 2 summarises all four domain instantiations. The routing code (`DivergenceRouter`, `AdapterRouter`, `DHardQueue`, `RoboticsDMN`) is byte-for-byte identical across all four domains. Empirical evidence for the domain-isomorphism claim in the vision-language setting is provided in [2]; robotics, finance, and CRISPR instantiations extend the architectural claim to physical sensor domains.

Domain	Stream A	Stream B	Failure classes
CRISPR screening	DNA structure (DNABERT-2)	Patient RNA profile (GeneJEPA)	<code>pooled_screen</code> , <code>genomic_structure</code>
Quantitative finance	Fundamental analysis	Market signals	<code>market_regime</code> , <code>structural</code>
Aviation sensor routing	Radar altimeter	Pitot tube	<code>weather_induced</code> , <code>hardware_struct</code>
<b>Humanoid robotics</b>	<b>Vision (camera)</b>	<b>Proprioception (IMU + joints)</b>	<b><code>environmental_transient</code>, <code>hardware_structural</code></b>

Table 2: Domain isomorphism across four L  r-JEPA instantiations. The temporal decay formula  $W = \exp(-\lambda \cdot \Delta t)$ , the identification/correction trust asymmetry (Definition 1), and the V1–V6 routing invariants are identical across all four domains. The  $\lambda$  constants and failure-class labels are the only domain-specific parameters.

## 8 Limitations and Future Work

**No empirical results.** This paper establishes the theoretical mapping and constitutes prior art. Empirical validation—including AIA Experiment 3 (SIGREG routing amplification,  $\rho > 1.15$ ) and a hardware evaluation on a physical humanoid platform—is deferred to future work.

**Real-time constraints.** System 1 centroid matching operates in sub-millisecond time (pure NumPy, no GPU). System 2 LORA injection loads a `.pt` file from disk—millisecond-range



latency on flash storage. For balance-recovery reflexes (ice slip,  $< 10$  ms response required) the router should rely on System 1 only. System 2 is appropriate for slower adaptation cycles: gait compensation after motor degradation detection, task re-planning after persistent sensor disagreement.

**Three-stream generalisation.** The current architecture routes two streams. Tactile sensors constitute a natural third stream, as would audio (for acoustic sensing of terrain or obstacles). Extending the V1–V6 contract to  $N$ -stream routing is an open problem identified in [1]; the D hard curriculum and consolidation cycle generalise straightforwardly but the routing decision function requires a multi-stream generalisation of Equation 1.

**Encoder training.** The encoders in the reference implementation are stub projection heads. For deployment on physical hardware, Stream A should be replaced with a pre-trained visual backbone (CLIP, DINOv2, or R3M) and Stream B with a proprioceptive model trained on robot motion capture data. The routing layer and DMN consolidation cycle are encoder-agnostic and require no modification.

**Threshold calibration.** The routing thresholds  $\tau_{\text{high}} = 0.60$ ,  $\tau_{\text{low}} = 0.25$ , and  $\delta = 0.35$  are initialised from the aviation domain. Calibration against physical sensor data—analogous to the DepMap CERES calibration in Snath Locus [3]—is required for deployment.

## 9 Conclusion

We have formalised the application of the V1–V6 cognitive routing contract to humanoid robotics, producing the fourth domain instantiation of the Lár-JEPA architecture. The key contributions are: (i) the M1–M3 sensor-to-stream mapping (Section 3); (ii) robotics-specific temporal decay constants for three failure classes (Section 4); (iii) a precise statement of the routing-vs-actuation safety boundary (Section 5); and (iv) the swarm DMN mechanism with SIGREG integration (Section 6).

The reference implementation at <https://github.com/snath-ai/snath-robotics> is architecturally complete and passes all regression tests. The routing code is byte-for-byte identical to the three prior domain instantiations, providing the fourth proof of domain isomorphism.

**Acknowledgements.** Developed on personal hardware, outside employment, prior to any employment commencement. RFC 3161 genesis certificate v6.2: Jun 2 21:27:09 2026 GMT (FreeTSA.org). Published: [doi:10.5281/zenodo.20517446](https://doi.org/10.5281/zenodo.20517446). This work was conducted independently under the Snath AI Open Source Research Initiative.

## References

- [1] A. V. Sajeev. Universal cognitive routing: A ten-abstract-base-class specification for domain-agnostic agent execution. *Zenodo*, 2026. [doi:10.5281/zenodo.20278775](https://doi.org/10.5281/zenodo.20278775)
- [2] A. V. Sajeev. Divergence is not noise: Multi-stream routing without modal fusion and the safety-learning equivalence. *Zenodo*, 2026. [doi:10.5281/zenodo.20278781](https://doi.org/10.5281/zenodo.20278781)
- [3] A. V. Sajeev. Architecture is all you need: Pre-registration and protocol for empirical validation of the Lár training loop. *Zenodo*, 2026. [doi:10.5281/zenodo.20419182](https://doi.org/10.5281/zenodo.20419182)
- [4] Snath AI. Snath Aviation: V1–V6 cognitive routing for aviation sensor streams. *GitHub*, 2026. <https://github.com/snath-ai/snath-aviation>



- [5] Snath AI. Snath Basis: V1–V6 cognitive routing for quantitative finance. *GitHub*, 2026. <https://github.com/snath-ai/snath-basis>
- [6] A. Radford et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
- [7] M. Oquab et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024.
- [8] S. Nair et al. R3M: A universal visual representation for robot manipulation. In *CoRL*, 2022.
- [9] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. [arXiv:2106.09685](https://arxiv.org/abs/2106.09685)
- [10] J. Ba, G. E. Hinton, V. Mnih, J. Z. Leibo, C. Ionescu. Using fast weights to attend to the recent past. In *NeurIPS*, 2016, pp. 4331–4339.
- [11] A. Brohan et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023. [arXiv:2307.15818](https://arxiv.org/abs/2307.15818)