Crimson Publishers
Wings to the Research

Mini Review

Brain-Inspired Computing for Embodied Intelligence: Why Neuromorphic Methods Belong in Next-Generation Robotics

Tianhao Zhang*

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China

Abstract

Brain-inspired computing is becoming increasingly relevant to robotics because autonomous machines must operate under strict limits of power, latency, memory and uncertainty. This opinion-style mini review argues that neuromorphic computing, event-driven sensing and spiking neural networks offer a principled route toward embodied intelligence that is more compatible with real-world robotic deployment than purely cloud-centered or over-parameterized approaches. We summarize why classically trained deep models still struggle in low-power, low-latency settings, discuss how neuromorphic methods can improve perception-action loops and outline the remaining barriers, including training stability, benchmarking, hardware-software co-design and safety validation. We contend that the most promising near-term path is not to replace main-stream artificial intelligence, but to integrate brain-inspired modules into robotic systems that require continuous sensing, adaptive control and energy-efficient autonomy.

Keywords: Neuromorphic computing; Embodied intelligence; Event-based sensing; Edge AI

ISSN: 2832-4463

***Corresponding author:** Tianhao Zhang, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China

Submission: March 31, 2026
Published: June 09, 2026

Volume 5- Issue 3

How to cite this article: Tianhao Zhang*. Brain-Inspired Computing for Embodied Intelligence: Why Neuromorphic Methods Belong in Next-Generation Robotics Brain-Inspired Computing for Embodied Intelligence: Why Neuromorphic Methods Belong in Next-Generation Robotics. COJ Rob Artificial Intel. 5(3). COJRA. 000615. 2026.
DOI: 10.31031/COJRA.2026.05.000615

Introduction

Robotics and artificial intelligence are converging toward a common goal: building embodied agents that can perceive, decide and act robustly in dynamic environments. Yet many current robotic pipelines still depend on compute-heavy deep learning stacks designed for abundant data and powerful hardware [1]. This mismatch becomes visible in edge robots, mobile manipulators, drones and wearable systems, where energy, bandwidth and reaction time are tightly constrained. In contrast, biological nervous systems achieve remarkable efficiency through sparse, event-driven and adaptive computation. Brain-inspired computing therefore deserves renewed attention not as a metaphor, but as an engineering framework for robotic autonomy.

This topic is important because the next generation of robots will not operate only as offline perception systems or cloud-connected executors. They will be expected to respond continuously to changing scenes, uncertain contact, intermittent communication and safety-critical constraints while using limited on-board resources. The central limitation of many conventional artificial intelligence pipelines is therefore not only predictive accuracy, but the gap between dense, frame-based, clock-driven computation and the sparse, asynchronous and adaptive structure of real-world sensorimotor interaction [2,3]. Recent reviews of event-based vision and neuromorphic robotics show that this gap is especially visible in high-speed navigation, tactile interaction, manipulation and mobile autonomy, where latency and energy are part of the task rather than secondary engineering details [3,4].

This mini review is organized around a simple question-based outlook: why do embodied robots need brain-inspired computation, what components of neuromorphic computing are most relevant, how can these components be connected to classical robotic pipelines and which evaluation criteria should guide future deployment? This framing is intended to clarify the motivation of the field before discussing architectures and bottlenecks. The article therefore treats neuromorphic computing not as a complete replacement for mainstream deep learning, but as a complementary route for building robotic systems that sense, learn and act under realistic physical constraints.

Literature Review and Research Outlook

Neuromorphic computing refers broadly to hardware and algorithms that emulate neurobiological principles such as spike-based communication, sparse coding, local memory-compute integration and asynchronous processing [5-7]. These properties align naturally with the needs of robotic systems. First, event-driven computation can reduce redundant processing and therefore lower energy consumption. Second, spike-based temporal coding is well suited to reactive control and continuous sensorimotor loops. Third, local adaptation may support operation in environments where connectivity is intermittent and retraining in the cloud is impractical.

Event-based cameras provide a concrete example. Unlike frame-based sensors, they report only pixel-level intensity changes, offering high temporal resolution, low motion blur and reduced data flow [3]. For fast navigation, tactile exploration and agile manipulation, these characteristics can improve responsiveness while keeping computation tractable. Combined with Spiking Neural Networks (SNNs), event streams can be processed in a manner closer to the physics of the task and the timing of action.

The central systems-level argument is summarized in Figure 1. Traditional pipelines often separate sensing, inference, planning and control into modules exchanging dense representations at fixed rates. Neuromorphic approaches encourage a different architecture: sparse sensing, asynchronous updates and direct coupling between salient events and motor decisions. This can benefit obstacle avoidance, slip detection in grasping, adaptive locomotion and anomaly monitoring in industrial robots.

The research outlook emerging from this literature is therefore not a single algorithmic problem, but a systems problem. Future work must identify which part of the embodied loop should be spike-based, which part should remain dense and semantic and how the two modes should exchange information without losing timing advantages. This question-driven view also helps readers understand why benchmarking and deployment validation are as important as model design: the value of a neuromorphic module depends on whether it improves the closed-loop robot under realistic energy, latency, robustness and safety requirements (Figure 1).

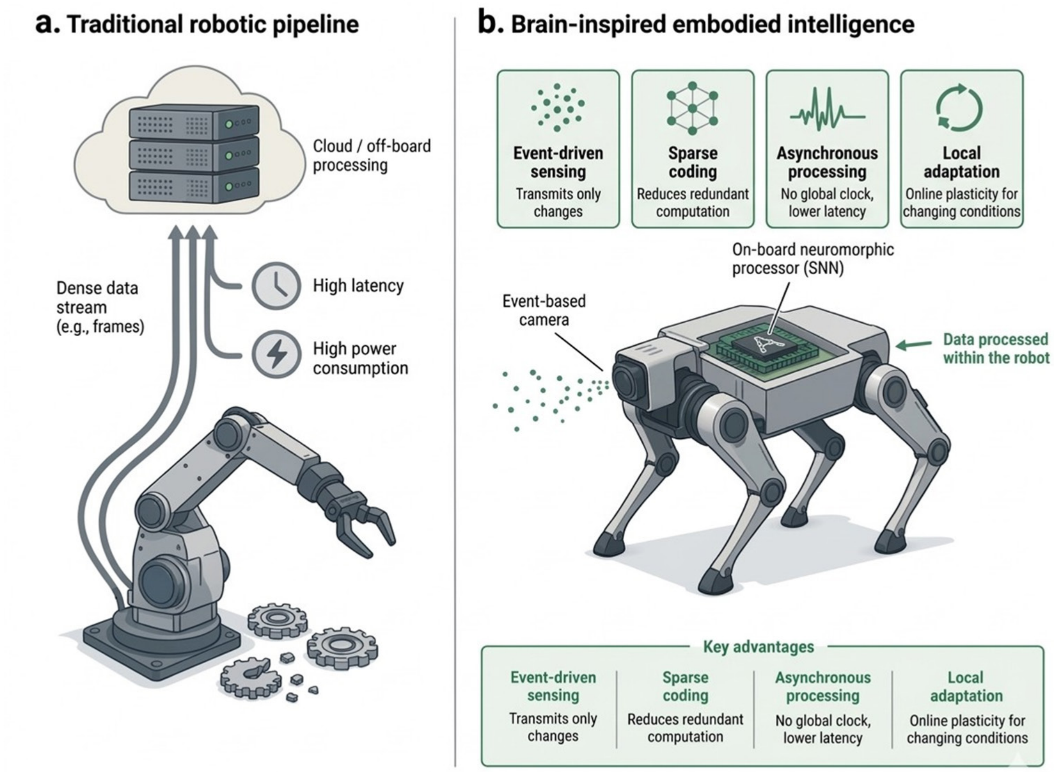

Figure 1: Comparison between a traditional cloud-heavy robotic pipeline and a brain-inspired embodied architecture emphasizing event- driven sensing, sparse coding, asynchronous processing, and local adaptation.

Methodological Framework: A Deployable Hybrid Architecture

Figure 2 suggests that the most realistic near-term path is hybrid rather than purist [8]. Fully neuromorphic robots remain a long-term goal, but near-term deployment is more likely to come from layered systems with different computational roles. Neuromorphic front ends can support fast perception, temporal filtering, closed-loop control and local policy refinement, while conventional deep models remain useful for semantic abstraction, multimodal context, long-horizon reasoning, and world knowledge [9]. This pragmatic division preserves edge responsiveness and energy efficiency while retaining the representational richness needed for high-level interpretation.

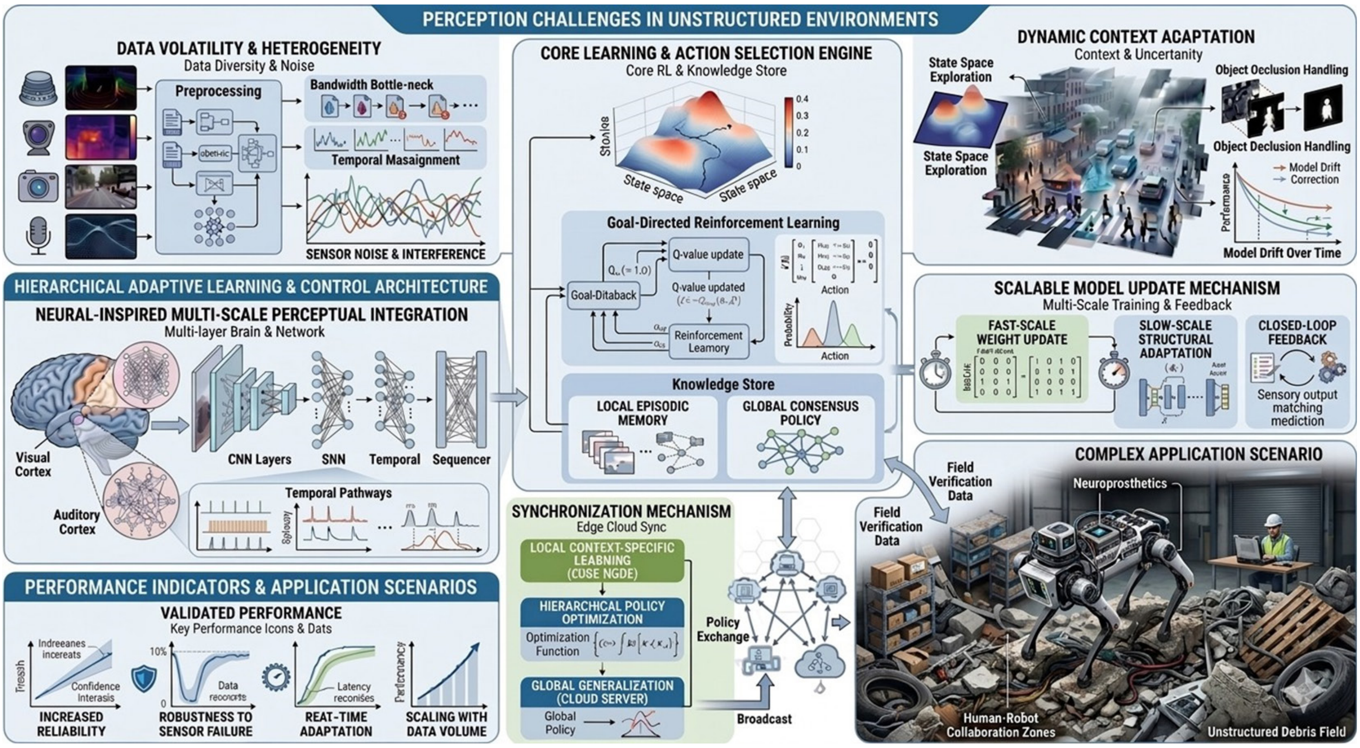

Figure 2: A deployment-oriented blueprint for brain-inspired embodied intelligence: heterogeneous sensing and preprocessing feed a neural- inspired multi-scale perceptual core, which connects to reinforcement learning, knowledge storage, hierarchical adaptation, synchronization, and closed-loop feedback for operation in unstructured environments.

In practice, this hybrid view needs simple interfaces: neuromorphic modules can transform asynchronous event streams into sparse temporal features or reflex-level action proposals, while conventional modules provide semantic context, task goals, memory retrieval and safety constraints.

A practical interface can be described in three layers. At the sensing layer, event cameras, tactile arrays and proprioceptive sensors produce asynchronous streams that are filtered locally to suppress noise and preserve timing. At the neural-inspired processing layer, SNNs or sparse temporal modules convert these streams into low-latency features, anomaly flags, or reflex-level control proposals. At the semantic and supervisory layer, classical learning modules, planners, or foundation-model components interpret the broader task context, impose safety constraints and decide when cloud-assisted updates are needed. This layered interface prevents neuromorphic computation from becoming an isolated hardware novelty and makes it part of a testable robotic architecture.

Recent work supports a layered logic spanning embodied SNNs, event-driven vision, neuromorphic manipulation and robotic control near sensors and actuators [9-12]. As in Figure 2, a deployable hybrid architecture can unite heterogeneous sensors, local episodic memory, adaptive policy updates, and cloud-assisted refinement in one closed loop. The key question is not replacement, but how to distribute computation across sensing, learning and action so each layer contributes where it is strongest. For robotics, this strategy offers a credible path from lab demonstrations to scalable, field-responsive systems that still benefit from broader machine-learning advances.

Results and Discussion: Applications, Evaluation, and Open Bottlenecks

The clearest opportunities lie in mobile robots, drones, industrial inspection, neuroprosthetic and human-robot interaction, where timely response and energy efficiency matter as much as predictive accuracy [1,13]. Broad adoption also requires stable training and careful front-end processing. Because spike generation is non-differentiable, SNN training usually relies on surrogate gradients and remains sensitive to temporal credit assignment and hyperparameter choice; a practical strategy is surrogate-gradient pretraining followed by reinforcement- learning fine-tuning. Benchmarks should report latency, joules per inference, robustness and deployment stability rather than accuracy alone, while denoising, temporal alignment and uncertainty-aware filtering can materially affect performance under noisy or safety-critical conditions. Thus, Figure 2 is not only an architecture map, but also a research agenda linking perception, learning, synchronization and real-world verification.

As the main contribution of this article, Figure 2 can be read as a deployment roadmap rather than only an illustration. For mobile robots and drones, the near-term value is rapid event-driven perception under motion blur, strong illumination change, and bandwidth limits. For manipulation and tactile exploration, spike-based temporal coding can help capture contact transitions, slip events and fast force changes that are easily diluted in frame-based pipelines. For industrial inspection and human-robot interaction, local adaptation and uncertainty-aware filtering can reduce unnecessary cloud dependence while preserving responsiveness in noisy environments. These examples show that the contribution is not a claim that every robotic function should be neuromorphic, but a principled allocation of computation according to timing, energy, and semantic requirements. Evaluation should therefore follow the same deployment logic. A useful benchmark should compare not only classification or detection accuracy, but also end-to-end response time, energy per decision, robustness under sensor noise, stability after distribution shift and failure behavior in closed-loop operation. In safety-critical robots, an architecture that is slightly less accurate in offline testing may be preferable if it produces earlier warnings, lower latency and more predictable degradation. Conversely, a high-performing neuromorphic module should not be considered deployable unless its interface with classical planning, memory and supervisory control is explicitly validated. This systems-level evaluation framework is essential for moving the field from isolated demonstrations to reliable embodied intelligence.

Several bottlenecks remain open. First, SNN training still lacks the maturity and reproducibility of conventional deep learning, especially when long temporal credit assignment and reinforcement-learning fine-tuning are involved. Second, event-driven sensors require careful preprocessing, temporal alignment, calibration and denoising; otherwise, the advantage of sparse sensing can be offset by unstable inputs. Third, hardware-software co-design is still fragmented across chips, simulators, datasets and robotic platforms. Finally, safety validation remains underdeveloped: neuromorphic robots must be tested not only for speed and efficiency, but also for interpretable failure modes, graceful fallback to conventional control, and stable behavior during online adaptation (Figure 2).

Conclusion

For next-generation robots, intelligence must be accurate, timely and energy-aware. Brain-inspired computing matters because it directly addresses this three-way requirement. The most persuasive near-term path is therefore selective integration: deploying neuromorphic sensing and spike-based computation where temporal sparsity and local autonomy are decisive, while retaining mainstream AI where dense representation learning remains superior. On that basis, neuromorphic methods should be viewed not as peripheral to robotics, but as an increasingly practical foundation for scalable embodied intelligence. Future use cases are likely to include agile autonomous navigation, low-power field robots, neuroprosthetic interfaces, tactile manipulation and safety-critical industrial inspection systems that require continuous local adaptation.

References

1. Chowdhury SS, Sharma D, Kosta A, Roy K (2025) Neuromorphic computing for robotic vision: Algorithms to hardware advances. Commun Eng 4(1): 152.
2. Roy K, Jaiswal A, Panda P (2019) Towards spike-based machine intelligence with neuromorphic computing. Nature 575(7784): 607-617.
3. Gallego G, Delbruck T, Orchard G, Bartolozzi C, Taba B, et al. (2022) Event-based vision: A survey. IEEE Trans Pattern Anal Mach Intell 44(1): 154-180.
4. Aitsam M, Davies S, Nuovo AD (2022) Neuromorphic computing for interactive robotics: A systematic review. IEEE Access 10: 122261-122279.
5. Marchisio A, Shafique M (2025) Neuromorphic computing for embodied intelligence in autonomous systems: Current trends, challenges, and future directions. 31st IEEE International Symposium on On-Line Testing and Robust System Design (IOLTS), Ischia, Italy, pp: 1-4.
6. Yik J, Van den Berghe K, den Blanken D, Bouhadjar Y, Fabre M, et al. (2025) The neurobench framework for benchmarking neuromorphic computing algorithms and systems. Nat Commun 16(1): 1545.
7. Muir DR, Sheik S (2025) The road to commercial success for neuromorphic technologies. Nat Commun 16: 3586.
8. Barbier T, Teulière C, Triesch J (2025) A spiking neural network for active efficient coding. Front Robot AI 11: 1435197.
9. Wang Z, Zhao J, Yang J, Wang, Y, Xiao, X, et al. (2025) Embodied neuromorphic control applied on a 7-DOF robotic manipulator. ArXiv.
10. Ahmadvand R, Sharif SS, Banad YM (2025) Neuromorphic robust framework for integrated estimation and control in dynamical systems using spiking neural networks. Sci Rep 15(1): 44753.
11. Zhang S, Zha F, Wang X, Li M, Guo W, et al. (2025) High-efficiency sparse convolution operator for event-based cameras. Front Neurorobot 19: 1537673.
12. Akcal U, Raikov IG, Gribkova ED, Choudhuri A, Kim SH, et al. (2025) LoCS-Net: Localizing convolutional spiking neural network for fast visual place recognition. Front Neurorobot 18: 1490267.
13. Zhu Z, Chen K, Lin W, Yan H, Liang L, et al. (2025) Recent advances in spike-based neural coding for tactile perception. Microsyst Nanoeng 11(1): 212.