SegMind

Segmenting and Explaining Robot Actions by Reasoning over Semantic Digital Twins

Sorin Arion
AICOR Institute of Artificial Intelligence
University of Bremen
Bremen, Germany
sorin@uni-bremen.de

Abdelrhman Bassiouny
AICOR Institute of Artificial Intelligence
University of Bremen
Bremen, Germany
bassioun@uni-bremen.de

Giang Nguyen
AICOR Institute of Artificial Intelligence
University of Bremen
Bremen, Germany
hoanggia@uni-bremen.de

Michael Beetz
AICOR Institute of Artificial Intelligence
University of Bremen
Bremen, Germany
beetz@cs.uni-bremen.de

***Abstract*—Safe and verifiable deployment of autonomous robots in open-ended environments requires causal, intentional understanding of robot behavior and bridging the gap between continuous sensorimotor data and symbolic reasoning for transparent interpretation and verification. We propose SegMind, which builds structured, queryable models of robot actions by simulating interactions in a Semantic Digital Twin (SDT) and performing continuous event perception over state trajectories. It detects physical events such as contact, lifting, placement, and manipulation, and composes them into higher-level actions that are reasoned over using Entity Query Language (EQL), producing explicit inference traces. Experiments on a TIAGo dual-arm kitchen task show accurate multi-stage segmentation under noise, precise event reconstruction, and interpretable, auditable behavior explanations.**

***Index Terms*—Event Segmentation, Explainability, Transparent Reasoning, Knowledge Representation and Reasoning, Episodic Memory.**

I. Introduction

The field of Physical AI is at a critical juncture. As intelligent systems transition from restricted industrial settings to dynamic social and domestic spaces, the requirements for safety, robustness, and trustworthiness become paramount [1]. Traditional safety mechanisms, such as force-limiting actuators or control barrier functions, are insufficient for agents that must make complex decisions under uncertainty. True safety in physical AI requires ”interpretability” and ”intent understanding” [1].

Consider a robot operating in a domestic kitchen. Its sensors produce a continuous stream of pose estimates, joint states, contact forces, and object positions. From this raw trajectory of observations, the robot must extract meaning: the gripper is now holding the milk box, the support relation to the counter has been lost, the fridge door is being pulled open. Without this capacity to parse low-level physical data into structured, semantically grounded representations [3] of ongoing actions and environment changes, the robot cannot verify whether its plan is succeeding, diagnose why it failed, or adapt its behavior accordingly. Meanwhile, Human cognition overcomes this by segmenting activity into discrete events [5], [6], a capability we aim to adapt for robotic systems.

We introduce **SegMind**, an open-source framework [23] that integrates these cognitive principles into a unified architecture for safe robot control. SegMind employs Event Segmentation Theory (EST) to detect event boundaries where the most significant physical or conceptual state changes occur [3], [6]. These events are used to automatically enrich episodic memories [8], [11] which are represented in SegMind as event logs that are accompanied by structured explanations for the detected events.

SegMind is built upon the **KRROOD** paradigm [2], using **Entity Query Language (EQL)** for declarative reasoning over a **Semantic Digital Twin (SDT)** which provides the world knowledge in the form of semantic annotations, kinematic structures, spatial, and geometric relations and properties.

II. Related Work

A. Event Segmentation in Robotics

Event segmentation is the process of parsing continuous experience into meaningful chunks [3]. Historically, this

has been addressed through three primary paradigms. *Rule-based* systems utilize manually defined criteria [4]. *Training-based* approaches employ deep learning models, such as LSTMs or Transformers, to achieve high-precision action recognition from large datasets. Recently, *LLM-based* and Vision-Language Model (VLM) frameworks have emerged, leveraging foundation models to segment robot motion from natural language descriptions or to perform zero-shot spatio-temporal reasoning [19], [20]. However, these statistical methods often lack the transparency required for safety-critical applications. SegMind bridges this gap by grounding segmentation in an explicit, queryable knowledge base [2].

B. Transparent Reasoning and Knowledge Representation

Safe Physical AI requires that a robot's belief state be synchronized with its execution environment [2]. Traditional Knowledge Representation and Reasoning (KR&R) tools often rely on external ontologies (OWL) and logic programming (Prolog), creating an "Object-Ontological Impedance Mismatch" [21], [22]. Recent frameworks like KRROOD address this by treating Python classes as native knowledge abstractions [2]. To interact with this knowledge, declarative languages like EQL provide a formal way to ask questions and add rules to this knowledge [2].

C. Failure Handling and Safety Verification

Current embodied AI methods frequently struggle with overconfident predictions that risk user safety [1]. Explainable failure handling requires systems that can perform causal diagnosis to generate structured explanations of *why* a failure occurred. SegMind contributes to this by generating explanation graphs that link high-level event labels to the underlying EQL queries and rules that triggered them.

III. Methodology

SegMind's event detection pipeline rests on two pillars: a *Semantic Digital Twin* (SDT) that provides a continuously updated geometric and symbolic model of the environment from real sensor data, and a set of *Event Detector Statecharts*, which rely on *Motion Statecharts* [15] that encode hierarchical event-detection logic as reactive state machines. Together, they transform raw pose streams into a certified symbolic trace of robot behavior.

A. Semantic Digital Twin

The Semantic Digital Twin is a environment model that couples geometric reasoning with semantic expressiveness. Formally, the environment at time t is represented as a tuple

$$\mathcal{E}(t) = (\mathcal{O}, \mathcal{G}, \mathcal{R}), \tag{1}$$

where $\mathcal{O} = \{o_1, \ldots, o_n\}$ is the set of objects, $\mathcal{G} = \{g_i\}$ assigns each object o_i a geometric model together with its current 6-DoF pose, and $\mathcal{R}$ is a directed scene graph whose typed edges encode spatial relationships. Each object carries semantic annotations linking it to its object class, functional role, and affordances (e.g., `Aperture` for openings that admit insertion), allowing purely geometric computations to produce semantically meaningful predicate values such as `Containment` and `Support` without any learned parameters.

SegMind operates on an instance of the SDT rather than directly on raw sensor streams. Pose updates are received through a standardized interface and applied to the twin's state, after which all reasoning is performed over the SDT. This state is then consumed by the Motion Statechart network described next.

B. Event Detection Architecture

SegMind uses *Event Detector Statecharts* as an execution graph where each node is a type of *Event Detector* that contains the detection logic implemented through EQL-Rules. On every state update, the network *ticks*: each node re-evaluates its preconditions against the current SDT state and, if satisfied, executes its detection logic. A node becomes active when its start precondition (a logical formula over other nodes' observation states) evaluates to true, forming a directed acyclic dependency graph that describes multi-step event detection without requiring the user to manually write detection logic.

C. Detector Class Hierarchy

All detectors in SegMind share a common base that provides infrastructure for object filtering, geometric relation queries, and event logging. Each detector receives the current world state, a shared detection context, and the list of objects to inspect for the current tick, and produces a list of events as output.

The shared detection context serves as the collective memory of the pipeline across all detectors within a tick. It accumulates all intermediate state required for hierarchical reasoning: the current contact graph between bodies, active support and containment relations, motion state, and the full event timeline, of already detected events. Because all detectors read from and write to the same context instance within a single tick, higher-level detectors can safely consume the results of lower-level ones without additional synchronization.

D. Event Hierarchy

SegMind organizes detected events into three levels, each building on the one below.

Atomic Events

Atomic events are computed directly from pairwise geometric queries on the SDT and require no prior event history. A *contact event* or its counterpart *loss of contact event* is fired when the bounding volumes of two bodies intersect or separate, subject to a "debounce counter" that suppresses spurious transitions caused by numerical boundary oscillation: a contact is only confirmed after N_{debounce} consecutive frames of intersection, and a loss only after N_{debounce} frames of separation. A *translation event* fires when an object's displacement over a sliding window of W frames exceeds a configurable threshold $\varepsilon_{\text{move}}$, and its counterpart *stop translation event* fires when the object falls back below it. Analogous *rotation events* track angular displacement via the geodesic distance between consecutive rotation matrices.

Spatial Events

Spatial events encode object-level relationships that require the SDT's scene-graph reasoning layer. A *support event* fires when one object is resting on top of another, determined by geometric overlap of their bounding projections along the gravity axis, and its counterpart *loss of support event* fires when that relationship no longer holds. *Containment events* fire when the center of mass of one object lies within the volumetric boundary of another. An *insertion event* is triggered when a body passes through a semantically annotated aperture and enters a containing volume. Spatial detectors use start conditions that require prior atomic events, avoiding unnecessary geometric computation.

Agent and Coarse Events

Agent events capture robot-specific manipulation semantics. A *holding event* fires when all finger bodies of a configured gripper group are simultaneously in contact with a tracked object, and its counterpart *loss of holding event* fires when a previously established hold is no longer supported by any gripper contact. A *lifting event* fires when a held object's z-position exceeds a baseline recorded at holding onset by a threshold ε_z. An *opening event* fires when the displacement vectors of a held handle and the gripping finger are correlated in both direction and magnitude over a sliding observation window, indicating that an articulated object is being actively moved.

Coarse events group lower-level events that occur together in time, using the shared context log. A *pick-up event* is detected when a translation event and a loss-of-support event co-occur on the same object within a configurable time window Δt. A *placing event* is detected when a stop-translation event and a support event co-occur within Δt.

E. Runtime Loop

The SegMind runtime proceeds as follows on each update cycle: (1) new pose estimates are received and applied to the SDT; (2) the statechart ticks, activating or deactivating detector nodes based on whether their preconditions are currently satisfied; (3) each RUNNING node queries the SDT or the context log and emits zero or more typed event objects; (4) emitted events are appended to the shared context log and broadcast to any registered listeners. The execution order within a tick follows the dependency order of the statechart, ensuring that atomic results are available to spatial detectors, and spatial results to agent and coarse detectors.

F. Event Explanation using EQL

When a robot executes an action plan, failures can manifest as either unexpected event detections or the absence of expected ones. For a planning library to recover, it must answer not just *what* happened but *why* an event was (or was not) detected.

SegMind uses the Entity Query Language (EQL) declarative queries and inference rules; these rules can be used to infer (detect) events. EQL records the full inference context of every inferred conclusion; these records are used to explain any conclusion through an EQL built-in function called *explainInference*, which takes as input the conclusion to be explained and produces an *InferenceExplanation* that is a built-in functionality in EQL. This explanation has multiple components. First, it has the rule that produced this inference. Second, it contains the exact bindings for all the rule nodes (variables and conditions) that existed in this specific inference, which also includes what conditions were evaluated and which were satisfied. Third, it contains the stack trace at the point the rule was invoked; this stack is clipped to the point where it gets outside of the code library (for example, when it goes into externally installed packages).

Furthermore, SegMind provides an *EventExplainer* interface that makes use of EQL's *InferenceExplanation* and extends it for the purposes of event explanation. These extensions are general-purpose introspective queries that provide useful information in the context of event detection. Events as discussed in III-D are hierarchical, so higher-level events are composed of one or more lower-level events. This suggests that a useful query would ask what other events were used to infer this event. Another useful query would be what relations must hold between these events. These relations are then a description of the inferred event. For the actual queries, check [23].

IV. EXPERIMENTS

A. Setup

We evaluate SegMind on a TIAGo [18] dual-arm robot manipulation episode simulated in MuJoCo [16] and recorded via Multiverse [17], which acts as the data

connector that streams synchronized body poses and joint states into SegMind's Semantic Digital Twin [23]. The episode comprises 6-DoF poses for all scene objects and joint states sampled at approximately 25 Hz in a domestic kitchen environment containing a refrigerator with an articulated door and a milk box as the primary manipulated object. The resulting data contains realistic interaction effects — intermittent contacts, pose oscillations, and object drift — that test whether SegMind's detectors remain stable under physical noise, making it a suitable proxy for real-world validation before hardware transfer. The full detector stack was deployed with debounce $N_{\text{debounce}} = 3$, translation window $W = 20$ frames and $\varepsilon_{\text{move}} = 0.01\,\text{m}$, lifting threshold $\varepsilon_z = 0.05\,\text{m}$, opening window of 10 frames with $\varepsilon_{\text{disp}} = 0.01\,\text{m}$ and angular threshold $0.3\,\text{rad}$, and coarse event window $\Delta t = 2\,\text{s}$. Although SegMind is designed to run online alongside robot action execution, we evaluate it here in an offline replay setting, feeding the recorded episode into the pipeline frame by frame. This allows for controlled and reproducible evaluation without requiring a running robot or an active simulation. The pipeline processed frames at approximately 16 Hz on a machine running Ubuntu 24.04, equipped with an Intel Core i7-13700, an NVIDIA RTX 4070, and 64 GB of RAM.

B. Results

The pipeline produced 66 events over an episode spanning approximately two minutes. The ground truth consists of four semantically significant manipulation events, two *holding events*, a *pick-up event*, and an *opening event*. These are manually annotated by a human observer watching the episode replay. The remaining 60 detected events are lower-level atomic and spatial events such as *contact*, *support*, and *translation* events that compose the annotated manipulation phases but were not individually labelled. Figure 1 shows the detected timeline against the ground truth over the 60–140 s interval where all key manipulation events occur, with all four ground truth events reproduced within a small temporal margin.

C. Discussion

These are preliminary results on a single episode, intended to demonstrate the feasibility of the approach rather than provide a comprehensive quantitative evaluation. The temporal alignment between detected and annotated events confirms that the hierarchical detector structure correctly assembles high-level manipulation phases from lower-level geometric evidence. Moreover, the explanation of the detected events provides all the needed information for downstream tasks like failure handling of action executions. The detection explanation of this experiment tells that *PickUpEvent* action was detected through the EQL rule in the *PickUpDetector*, in addition, it shows that

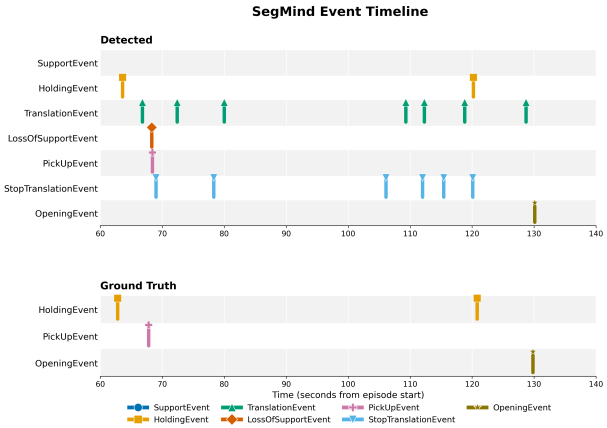

Fig. 1: Detected event timeline (top) vs. ground truth annotations (bottom) for the TIAGo kitchen episode [23].

the *TranslationEvent*, and *LossOfSupportEvent* were used to infer the *PickUpEvent*, and that they were constrained to be referencing the same object, and to have occurred within a specific duration threshold.

V. Conclusion and Future Work

This paper presented SegMind, a framework for transparent episode event segmentation that bridges the sensorimotor-symbolic gap in Physical AI. By coupling a Semantic Digital Twin (SDT) with hierarchical Motion Statecharts, the architecture enables real-time parsing of unstructured state parameters into verifiable, discrete events. Furthermore, leveraging the native object-oriented design of the KRROOD paradigm alongside the declarative expressiveness of the Entity Query Language (EQL) allows SegMind to generate high-fidelity inference traces that map high-level activities directly to their execution origins, physical variable bindings, and satisfied logical preconditions. Experimental validation under realistic contact dynamics demonstrates that the framework is able to detect interaction events, providing auditable decision trails crucial for runtime safety verification and intent explanation.

Future work will expand the framework along three primary trajectories. First, we will evaluate the deployment of the pipeline on physical robot hardware to validate its resilience against real-world sensor noise and mechanical uncertainty. Second, we will integrate learned vision modules such as deep neural networks, enabling SegMind to operate end-to-end on raw RGB-D streams. Third, we will link the introspective explanation interface directly to the closed-loop recovery routines of live task planners.

REFERENCES

[1] Workshop Committee, "Workshop on Safe Physical AI," 2025. [Online]. Available: https://safephysicalai.github.io/. [Accessed: 10-May-2026].

[2] A. Bassiouny, T. Schierenbeck, S. Arion, B. Alt, N. Vasantakumaar, G. Nguyen, and M. Beetz, "Implementing Knowledge Representation and Reasoning with Object Oriented Design," in *Proceedings of the 2026 International Joint Conference on Artificial Intelligence (IJCAI)*, 2026.

[3] J. M. Zacks, N. K. Speer, K. M. Swallow, T. S. Braver, and J. R. Reynolds, "Event segmentation," *Psychological Bulletin*, vol. 133, no. 2, pp. 273–293, 2007.

[4] J. M. Zacks, "Using movement and intentions to understand simple eventsv*Cognitive Science*, vol. 28, no. 6, pp. 979-1008, 2004.

[5] J. M. Zacks and B. Tversky, "Event structure in perception and conception," *Psychological Bulletin*, vol. 127, no. 1, pp. 3–21, 2001.

[6] C. A. Kurby and J. M. Zacks, "Segmentation in the perception and memory of events," *Trends in Cognitive Sciences*, vol. 12, no. 2, pp. 72–79, 2008.

[7] J. R. Reynolds, J. M. Zacks, and T. S. Braver, "A computational model of event segmentation from perceptual prediction," *Cognitive Science*, vol. 31, no. 4, pp. 613–639, 2007.

[8] S. Jockel, D. Westhoff, and J. Zhang, "Towards a high-level episodic robot memory," in *Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics*, 2008, pp. 1075–1081.

[9] M. Kümpel, C. A. Mueller, and M. Beetz, "Semantic Digital Twins for Retail Logistics," in *Dynamics in Logistics*, Springer, 2021, pp. 129–153.

[10] E. Tulving, "Episodic and semantic memory," in *Organization of Memory*, E. Tulving and W. Donaldson, Eds. New York: Academic Press, 1972, pp. 381–403.

[11] E. Tulving, *Elements of Episodic Memory*. Oxford, UK: Clarendon Press, 1983.

[12] A. Bannat et al., "Artificial Cognition in Production Systems," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, pp. 148–174, 2011.

[13] P. Compton and B. Jansen, "Knowledge elicitation and experience," in *Proceedings of the Second Australian Joint Conference on Artificial Intelligence*, 1988, pp. 363–377.

[14] B. Kang and P. Compton, "Knowledge acquisition and maintenance with ripple down rules," *Proceedings of the 1994 Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop*, 1994.

[15] S. Stelter, "A Robot-Agnostic Kinematic Control Framework: Task Composition via Motion Statecharts and Linear Model Predictive Control," University of Bremen, Feb. 2025. [Online]. Available: https://media.suub.uni-bremen.de/handle/elib/8862

[16] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033. DOI: 10.1109/IROS.2012.6386109

[17] G. Nguyen and M. Beetz, "Multiverse," Dec. 2024. [Online]. Available: https://github.com/Multiverse-Framework/Multiverse. DOI: 10.5281/zenodo.14035537

[18] P. Pages, J. Marchetti, and F. Ferro, "TIAGo: The Modular Robot that Adapts to Different Research Needs," in *International Workshop on Robot Modularity, IROS*, 2016.

[19] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models," in *Proceedings of the 7th Conference on Robot Learning*, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229, *Proceedings of Machine Learning Research*. PMLR, Nov. 2023, pp. 540–562.

[20] M. Ahn et al., "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," in *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2022, pp. 1–16.

[21] C. Ireland and D. Bowers, "Object-Relational Impedance Mismatch in the 21st Century," in *Proceedings of the 2015 International Conference on Software Engineering and Knowledge Engineering (SEKE)*, 2015, pp. 466–471.

[22] M. Ledvinka and P. Křemen, "A comparison of object-triple mapping libraries," *Semantic Web*, vol. 11, no. 3, pp. 483–524, 2020.

[23] CRAM-Organization, *SegMind*. Available: https://github.com/sorinar329/cognitive_robot_abstract_machine/blob/safety_ai_ws/segmind/demo/tiago_demo.py, 2026.