DOI: 10.34749/3061-1466.2026.13

KINESIS: A Schema-Driven Motion Capturing, Management and Augmentation Framework

Günther Hutter
Chair of Cyberphysical Systems
Technical University of Leoben
8700 Leoben, Austria
guenther.hutter@unileoben.ac.at

Abstract

Motion capture systems are widely used to generate training data for machine learning and robotics applications. However, existing workflows often rely on fragmented toolchains, ad-hoc preprocessing scripts, and inconsistent data representations, which complicates reproducibility and rapid experimentation.

We present Kinesis, an API-first schema-driven framework for capturing, validating, inspecting, and augmenting motion data based on structured 3D keypoints. The framework enforces consistent dataset structure through declarative skeletal schemas, enables interactive inspection through synchronized tabular and 3D visualization, and provides schema-aware augmentation operators for spatial and temporal transformations.

To demonstrate practical applicability, we implement a WebXR-based hand gesture capture pipeline using consumer-grade hardware (Meta Quest 3). While this setup serves as a concrete example, the framework is hardware-agnostic and extensible to other pose estimation sources. By unifying capture, validation, inspection, and augmentation in a single system, Kinesis provides a reproducible foundation for motion-based machine learning workflows and rapid prototyping in XR and robotics scenarios.

1 Introduction

Human motion understanding plays a central role in applications such as extended reality (XR), gesture-based interaction, and human–robot collaboration Feith and Rückert [2024]. Modern pose estimation systems produce temporally structured streams of 2D or 3D keypoints that can be used for recognition, control, and behavior modeling tasks.

Despite these advances, transforming raw motion data into reliable machine learning datasets remains challenging. Motion recordings often differ in structure, coordinate systems, and feature ordering, requiring substantial preprocessing before they can be used for training. In many workflows, these steps are implemented as ad-hoc scripts or tightly coupled preprocessing pipelines, which limits reproducibility and complicates experimentation.

These issues are particularly pronounced in few-shot learning scenarios, where only small numbers of recordings are available and systematic data augmentation becomes essential. Without a consistent structural representation of motion data, augmentation and dataset management become difficult to generalize across tasks.

To address these challenges, we introduce Kinesis, a schema-driven framework for capturing, organizing, validating, and augmenting motion datasets composed of named 3D keypoints. The framework

The Third Austrian Symposium on AI, Robotics, and Vision (AIROV26).

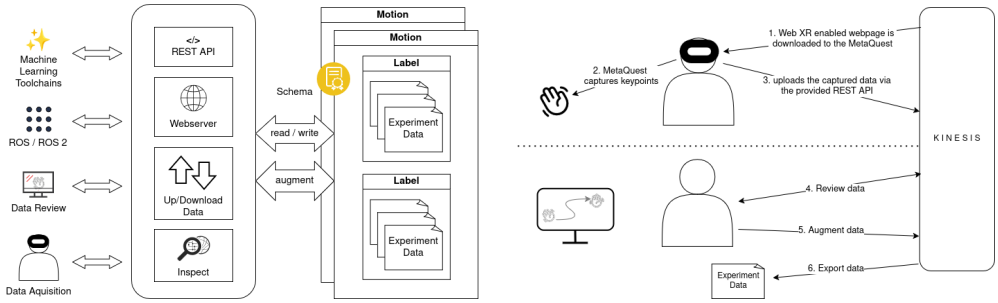


Figure 1: Overview of the Kinesis framework. Left: system architecture exposing capture, inspection, and augmentation functionality via a REST API, with motion data organized as schema-aligned labeled sequences. Right: example workflow illustrating WebXR-based motion capture, schema validation, dataset storage, inspection, augmentation, and export for downstream machine learning tasks.

enforces structural consistency through declarative skeletal schemas and integrates motion capture, dataset management, visualization, and augmentation within a unified API-based architecture.

The main contributions of this work are: (i)a schema-driven representation for structured motion datasets, (ii) a modular augmentation pipeline for motion learning scenarios, (iii) an API-based system for dataset management and reproducible experimentation, and (iv) a demonstration of the framework in a WebXR-based hand gesture acquisition pipeline using consumer hardware.

2 Related Work

Motion capture pipelines typically rely on specialized storage formats originating from animation or biomechanics communities. Formats such as BVH or FBX encode articulated body structures as hierarchical joint trees and are widely used in graphics pipelines of Wisconsin Graphics Group [1999], Autodesk Inc. [2020]. Marker-based formats such as C3D provide high temporal precision for laboratory motion capture systems C3D.org [2026], Müller et al. [2007].

In contrast, machine learning pipelines frequently rely on tensor-oriented storage formats such as HDF5, NPZ, or Apache Arrow, which are optimized for numerical data processing and efficient I/O Folk et al. [2011], Hopsworks Team [2019], Apache Software Foundation [2026]. While these formats provide efficient storage, they typically omit explicit structural metadata describing joint relationships or skeletal topology.

Existing pose estimation toolchains primarily focus on extracting keypoints rather than supporting the full dataset lifecycle including validation, inspection, and augmentation. As a result, developers often construct custom processing layers that transform raw motion streams into consistent training datasets.

Data augmentation has been shown to improve robustness in motion-based learning systems by introducing invariances to spatial transformations, temporal distortions, and sensor noise Um et al. [2017], Wang et al. [2019]. However, augmentation is frequently implemented as task-specific preprocessing rather than integrated into dataset management workflows.

Kinesis addresses these limitations by combining tensor-based motion storage with explicit schema-level structural metadata and schema-aware augmentation operators within a unified framework.

3 Kinesis Framework

Kinesis is implemented in Python and consists of four tightly integrated components for motion data storage, validation, inspection, and augmentation.

Storage and Validation Kinesis represents motion as temporally ordered trajectories of named 3D keypoints. Each recording is stored as a tabular dataset in which rows correspond to timestamps and

columns contain the coordinates of predefined joints or markers. Structural consistency is defined and validated through a schema, specified in Graphviz DOT [1] format, that describes expected features, connectivity, and optional visualization metadata.

Inspection The framework provides an integrated inspection engine for synchronized tabular and 3D visualization of motion data. Because inspection operates directly on the tensor representation and its associated schema metadata, no intermediate format conversion is required.

Augmentation Augmentations are defined as schema-aware transformations over motion tensors $X \in \mathbb{R}^{T \times J \times 3}$. Kinesis distinguishes four augmentation scopes: (i) point-level perturbations applied to individual tensor elements, (ii) frame-level transformations applied consistently to all joints in one time step, (iii) sequence-level transformations applied to the full recording, and (iv) temporal augmentations that modify sampling along the time axis.

This abstraction allows augmentations to be executed dynamically during loading or persistently stored as derived datasets while preserving structural guarantees.

Application Integration Kinesis exposes its functionality through a lightweight REST API and can be integrated with external capture systems or preprocessing services. This design supports rapid prototyping while maintaining compatibility with the internal tensor-based representation.

4 Example Application

To demonstrate the framework, we implemented a WebXR-based hand gesture acquisition pipeline using a Meta Quest 3 device. Hand joint positions are captured in the browser and transmitted to the Kinesis backend, where they are validated against a predefined schema and stored in a structured representation.

The integrated web interface supports dataset browsing, replay, and augmentation. Captured sequences can be extended through schema-aware transformations and exported for downstream machine learning experiments without additional format conversion.

For hand gesture capture, the schema defines joint names, connectivity, and column mappings. Uploaded datasets are required to contain a timestamp column and three spatial coordinates per joint. Missing required columns trigger validation errors, while valid inputs are normalized to the schema-defined column order.

5 Results & Conclusion

This work presented Kinesis, a schema-driven framework for motion capture, dataset management, validation, inspection, and augmentation. By combining tensor-based motion storage with explicit structural metadata, the framework reduces representational inconsistencies between capture pipelines and downstream machine learning systems.

The WebXR-based hand gesture acquisition example demonstrates that consumer-grade XR hardware can be integrated into a reproducible motion learning workflow with minimal preprocessing effort.

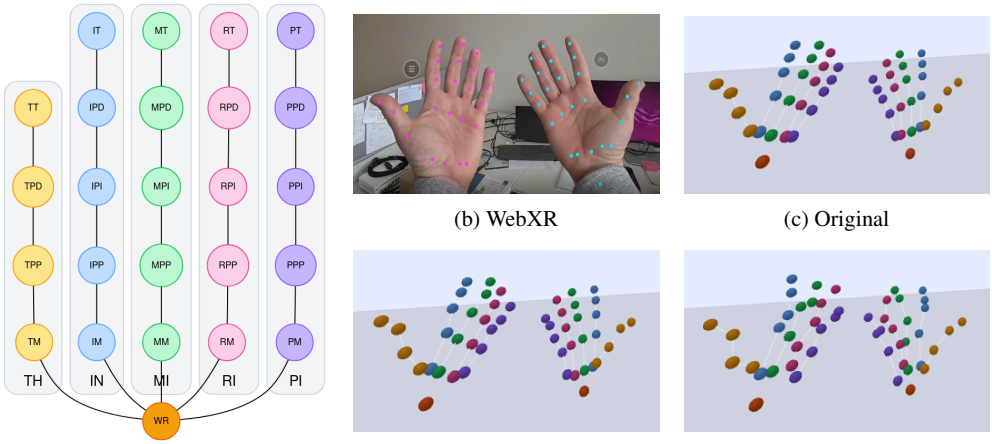
Future work will focus on scalable storage backends, extended augmentation strategies, tighter integration with training pipelines, and deployment-oriented interfaces for real-time inference and robotics applications.

References

Apache Software Foundation. Apache arrow vs parquet: Columnar formats for analytics, 2026. URL https://arrow.apache.org/docs/format/Columnar.html. Accessed: 2026-03-02.

Autodesk Inc. Autodesk fbx file format specification. https://help.autodesk.com/view/FBX/ENU/, 2020. Accessed: 2026-03-02.

[1] https://graphviz.org/doc/info/lang.html



(a) Schema

(b) WebXR

(c) Original

(d) 2% Noise

(e) 5% Noise

Figure 2: Schema-driven hand motion processing in Kinesis. Left: graph-based schema representation of the right-hand skeleton defining joint structure and connectivity. Top-right: real-time hand joint capture via a WebXR interface and replay of the detected pose in the integrated 3D viewer. Bottom-right: noise-based data augmentation with increasing perturbation magnitude while preserving the underlying skeletal topology enforced by the schema.

C3D.org. Specification - the c3d file format, 2026. URL https://www.c3d.org/HTML/Documents/specification.htm. Accessed: 2026-03-02.

Nikolaus Feith and Elmar Rückert. Advancing interactive robot learning: A user interface leveraging mixed reality and dual quaternions. In *IEEE International Conference on Ubiquitous Robots (UR 2024)*, pages 21–26. IEEE, 2024. doi: 10.1109/UR61395.2024.10597452. URL https://doi.org/10.1109/UR61395.2024.10597452.

Mike Folk, Gerd Heber, Quincey Koziol, Elena Pourmal, and Dana Robinson. An overview of the hdf5 technology suite and its applications. In *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*, pages 36–47, 2011. doi: 10.1145/1966895.1966900.

Hopsworks Team. Guide to file formats for machine learning. *Hopsworks Blog*, 2019. URL https://www.hopsworks.ai/post/guide-to-file-formats-for-machine-learning. Accessed: 2026-03-02.

Meinard Müller, Tido Röder, Michael Clausen, Bernd Eberhardt, Bernhard Krüger, and Andreas Weber. Documentation mocap database hdm05. In *Technical Report*. Max Planck Institute for Informatics, 2007. URL https://resources.mpi-inf.mpg.de/HDM05/07_MuRoClEbKrWe_HDM05.pdf.

University of Wisconsin Graphics Group. Biovision bvh format specification, 1999. URL https://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html. Accessed: 2026-03-02.

Terry T. Um, Franz M. J. Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction (ICMI)*, pages 216–220. ACM, 2017.

Lei Wang, Yonghong Nie, and Yuexian Li. Adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1116–1125, 2019.