KiRoSe: Using Kinematic Data for Efficient Segmentation of Robot Demonstrations

Anonymous authors
Paper under double-blind review

Abstract

Segmenting robot demonstration data into disjoint temporal and semantic subsegments is valuable for learning, cleaning, retrieval, and analysis. Segmentation is often performed manually in a tedious and error-prone process. Multimodal Large Language Models (MLLMs) have the potential for automating this process, but directly ingesting raw multi-camera video streams into MLLMs can be costly and can quickly exhaust context windows. In this paper, we introduce *KiRoSe* (Kinematics for Robot Segmentation), a novel framework that focuses on the companion stream of robot kinematic data. Inspired by the principle of classical entropy coding, such as Huffman coding, where frequently occurring symbols are represented more compactly, we propose Sparse Motion Tokens (SMTs), a discrete kinematic vocabulary designed to compactly represent recurring, temporally localized motion patterns, which we refer to as *kinematic events*. SMTs are learned by encoding kinematic trajectories with temporal-segment supervision and quantizing the resulting features. We use increasingly available robot demonstration subtask annotations to supervise an open language model to interpret the resulting token sequences. Evaluation against human-labeled segmentation shows that KiRoSe can substantially outperform baseline segmentation methods in F1 quality and transfers zero-shot to unseen robot embodiments and ego-centric human motion, while preserving the language model's open-vocabulary knowledge. Beyond segmentation, KiRoSe tracks task progress on an unseen robot and detects when an action is being reversed.

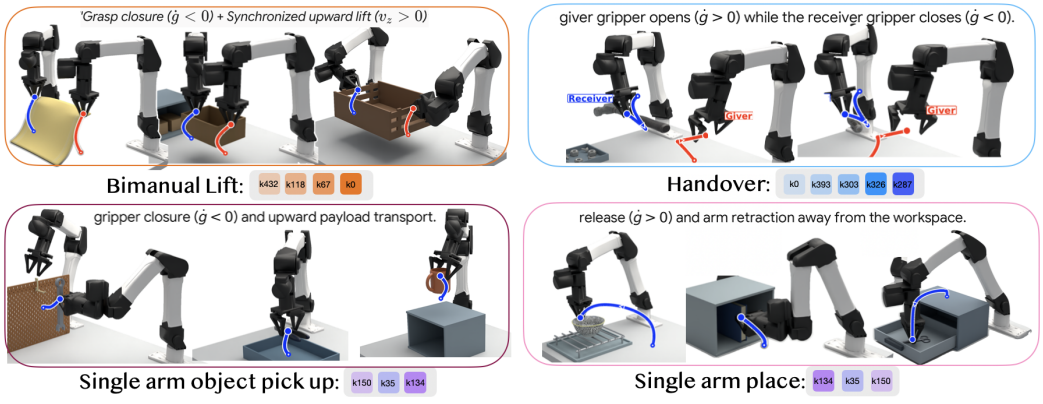


Figure 1: **Encoding Kinematic Events into Sparse Motion Tokens.** Across diverse manipulation tasks and objects, continuous robot motion decomposes into recurring, temporally localized patterns termed *kinematic events* (e.g., bimanual lift, handover, pick, place). Despite visual and spatial variations, end-effector trajectories within the same motion primitive share structural kinematic signatures. KiRoSe compresses kinematic events into a short, ordered sequence of N discrete Sparse Motion Tokens (colored blocks; shaded chronologically).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Large-scale robot demonstration datasets, such as Open X-Embodiment (Open X-Embodiment Collaboration, 2023), can grow to terabyte or even petabyte scale and span diverse robot embodiments and manipulation tasks. Beyond directly training Vision-Language-Action (VLA) models, understanding the structure within these trajectories is important for effectively using them. Robot demonstrations of "put the cup in the cabinet" and "put the book on the shelf," for example, share stages such as reaching, grasping, transporting, and placing, potentially mixed with pauses or corrective motions. We refer to recurring, temporally localized motion patterns underlying these stages as *kinematic events* (Figure 1). Identifying meaningful kinematic events supports demonstration generation through subtask reuse (Mandlekar et al., 2023), learning and sequencing reusable skills (Garrett et al., 2024), and retrieving relevant behaviors for policy adaptation (Memmel et al., 2024). Such intervals also provide useful units for inspecting demonstration quality, complementing approaches that improve policy learning through data filtering (Chen et al., 2025) or progress-based weighting (Chen et al., 2026b).

Manually annotating this structure is tedious and error-prone. Multimodal Large Language Models (MLLMs) offer a promising route to automate subtask annotation, but processing long, synchronized multi-camera videos can incur substantial inference costs and quickly exhaust their context windows. We observe that proprioceptive kinematics provide a compact and information-dense alternative to video: for example, gripper closure may indicate grasping, while a subsequent increase in end-effector velocity may indicate the transition from grasping to object transport (Figure 1). However, raw kinematic trajectories are high-frequency and high degree-of-freedom continuous time series without the discrete, semantically structured representation (e.g., directly convert a 1 minute bimanual robot trajectory at 30 Hz consumes approximately 500k text tokens).

In this paper, we introduce *KiRoSe* (Kinematics for Robot Segmentation), a framework that focuses on using compact kinematics representation to identify action and their boundaries. At its core are Sparse Motion Tokens (SMTs), a low-rate kinematic tokenizer for pretrained VLMs. Rather than asking a tokenizer to faithfully reconstruct every low-level state, we use the increasingly available subtask annotations in large-scale robot demonstration datasets as supervision for learning a sparse motion vocabulary that preserves information useful for language-model reasoning. We first encode high-frequency kinematic trajectories into temporally contextual features and quantize them into a compact discrete vocabulary. To make these tokens interpretable by a language model, we add the SMT vocabulary to a pretrained language model and supervise the reader with trajectory segment annotations containing segment boundaries and natural-language action descriptions. The tokenizer remains fixed during this stage, so the language model learns to interpret the motion vocabulary rather than reshaping the underlying representation.

We evaluate KiRoSe on held-out trajectory segmentation, zero-shot cross-embodiment transfer, semantic description of unseen actions, and task-progress understanding. Results show that KiRoSe achieves a boundary F1 within 1 second of 0.63 vs. 0.42 for the strongest alternative kinematic tokenizer on 200 held-out YAM episodes, and, transferred zero-shot to an unseen AgiBot embodiment, a boundary F1 within 0.5 seconds of 0.61 vs. 0.30 for GPT-6 Astra. KiRoSe also retains the language model's open-vocabulary knowledge, producing task-appropriate words absent from its training annotations on unseen human tasks; when a subtask is played backward, KiRoSe tracks the reversal with a progress correlation of $\rho = -0.68$ on AgiBot, compared with $\rho = +0.88$ for the video-based ReWiND baseline (Zhang et al., 2025).

To summarize, we make the following contributions: (1) **Sparse Motion Tokens (SMT)**, a boundary-supervised, low-rate kinematic tokenizer that compresses high-frequency robot motion into a compact discrete vocabulary with semantic and temporal motion structure; (2) **KiRoSe**, an end-to-end framework that adapts an open language model to interpret SMT sequences with temporal markers, enabling trajectory segmentation, zero-shot cross-embodiment transfer, open-vocabulary action description, and task-progress understanding; (3) **Evaluation Data** suggests substantial performance improvement in segmentation quality over baseline and frontier VLMs.

2 RELATED WORK

Trajectory Segmentation for Robot Demonstrations. Trajectory segmentation identifies temporal boundaries between motion primitives, with roots in biomechanics and motor control (Sternad &

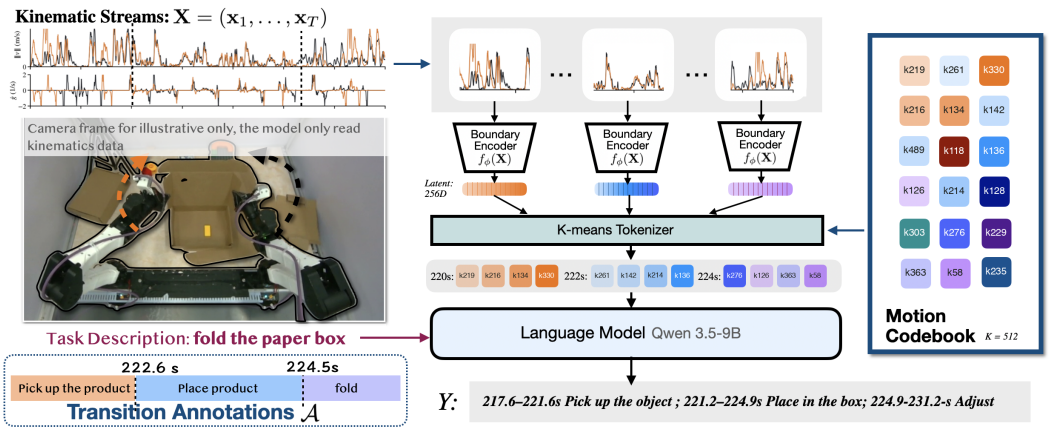


Figure 2: **KiRoSe Pipeline.** Given a robot kinematic stream and the task instruction "fold the paper box," KiRoSe encodes motion using a transition-supervised encoder, then temporally pools and quantizes its features into Sparse Motion Tokens with a learned codebook. The language model reads these tokens, time markers, and the instruction to output action descriptions with start and end times, such as "pick up the object," "place in the box," and "adjust." Transition annotations provide training supervision, while the motion tokenizer remains frozen during language-model adaptation.

Schaal, 1999; Viviani & Cenzato, 1985; Morasso, 1983). Probabilistic and hierarchical approaches discover these primitives from motion dynamics (Botvinick et al., 2009; Krishnan et al., 2017; 2019), often assuming a smooth, low-dimensional structure (Tanneberg et al., 2021; Iqbal et al., 2019). Modern robot learning datasets introduce additional challenges through diverse tasks, objects, and embodiments. RoboSegNet (Chen et al., 2026a) addresses multi-task trajectory segmentation by fusing visual and kinematic features to predict subtask boundaries. Building on this direction, KiRoSe studies how compact, transition-supervised kinematic tokens can support both temporal boundary prediction and natural-language action descriptions, using the task instruction to contextualize motion.

Kinematics Tokenization. Human motion tokenization and its integration with language models have been widely studied. Progress has been made in learning discrete vocabularies through vector quantization and training tokenizers to reconstruct continuous human motion and integrating the resulting tokens with language models for motion generation and understanding (Jiang et al., 2023; Zhou et al., 2024; Wu et al., 2025). Similar ideas have been explored in robotics, VQ-BeT uses residual vector quantization (Lee et al., 2024), FAST applies frequency-domain compression (Pertsch et al., 2025), and OAT learns ordered tokens supporting prefix reconstruction (Liu et al., 2026). These approaches prioritize reconstruction fidelity and efficient generation, whereas KiRoSe uses transition supervision to learn motion tokens that capture the precise kinematic events.

3 KiRoSe Pipeline

KiRoSe comprises two stages (Figure 2). First, we train a kinematic encoder with subtask-transition supervision, then temporally pool and quantize its features into SMTs. Second, we keep the motion tokenizer fixed and adapt a pretrained language model to predict segment boundaries and action descriptions from SMT sequences and the task instruction.

3.1 Transition Annotation Format

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ denote a robot kinematic stream, where $\mathbf{x}_t$ is the robot state at time step t. We annotate action transitions as

$$\mathcal{A} = \{(\tau_i, y_i^-, y_i^+)\}_{i=1}^N,$$

where N is the number of annotated transitions, τ_i is the timestamp of the i-th transition in seconds from the start of the trajectory, and y_i^- and y_i^+ are the semantic subtask labels immediately before and after the transition, respectively, with $y_i^- \neq y_i^+$. Each annotation therefore marks a temporal

boundary between two manipulation subtasks (as shown in Figure 2, left corner). Given a kinematic stream $\mathbf{X}$ and its task instruction, trajectory segmentation aims to identify temporal boundaries between subtasks and describe the action performed within each resulting segment.

3.2 Transition-Supervised Sparse Motion Vocabulary

Temporal Motion Encoding. Compressing a motion stream into discrete tokens should preserve the temporal structure of manipulation subtasks. We train a temporal encoder to capture motion context relevant to annotated subtask transitions. Denote temporal encoder as f_ϕ, parameterized by ϕ, maps $\mathbf{X}$ to a sequence of contextual features: $\mathbf{H} = f_\phi(\mathbf{X}) = (\mathbf{h}_1, \ldots, \mathbf{h}_T), \mathbf{h}_t \in \mathbb{R}^d$, where d is the feature dimension. The encoder uses residual temporal convolutions to aggregate neighboring motion states, allowing each feature $\mathbf{h}_t$ to represent local motion in its temporal context.

During training, we attach an auxiliary boundary-prediction head $b_\psi : \mathbb{R}^d \to \mathbb{R}$, parameterized by ψ, maps each feature to a scalar transition logit. The head predicts a transition probability at each time step: $p_t = \sigma(b_\psi(\mathbf{h}_t))$, where σ denotes the sigmoid function. To supervise these predictions, we convert the annotated transition timestamps $\{\tau_i\}_{i=1}^N$ into soft boundary targets $s_t \in [0, 1]$, which assign larger values to time steps near annotated transitions. We jointly optimize the encoder and boundary-prediction head using $\mathcal{L}_{\text{transition}}(\phi, \psi) = \frac{1}{T}\sum_{t=1}^{T} \ell_{\text{focal}}(p_t, s_t)$, where ℓ_{focal} is the focal binary cross-entropy loss. This objective encourages the motion features to retain information useful for locating subtask transitions. After training, we discard b_ψ and retain f_ϕ to produce continuous features for compression into discrete motion tokens.

Sparse Motion Tokens. High-frequency motion features contain substantial temporal redundancy. We compress the contextual features $\mathbf{H}$ into a low-rate sequence of discrete motion tokens through temporal pooling and vector quantization.

We freeze the temporal encoder f_ϕ and apply temporal pooling on a uniform grid to obtain a shorter feature sequence $\bar{\mathbf{H}} = (\bar{\mathbf{h}}_1, \ldots, \bar{\mathbf{h}}_M)$, where M is the number of pooled features and $\bar{\mathbf{h}}_j \in \mathbb{R}^d$. Each pooled feature is then quantized into a discrete motion token. We construct a motion vocabulary $\mathcal{C} = \{\mathbf{c}_k\}_{k=1}^K$, where $\mathbf{c}_k \in \mathbb{R}^d$ is a codebook centroid and K is the vocabulary size. Each pooled feature is assigned to its nearest centroid, producing a discrete token index $z_j = \arg\min_{k\in\{1,\ldots,K\}} \left\|\bar{\mathbf{h}}_j - \mathbf{c}_k\right\|_2^2$. The codebook is fitted using K-means on pooled features from the training trajectories only, with $K = 512$. Specifically, letting $\mathcal{P}_{\text{train}}$ denote the collection of training features, we minimize the quantization objective $\mathcal{L}_{\text{quant}}(\mathcal{C}) = \frac{1}{|\mathcal{P}_{\text{train}}|}\sum_{\mathbf{v}\in\mathcal{P}_{\text{train}}} \min_{k\in\{1,\ldots,K\}} \|\mathbf{v} - \mathbf{c}_k\|_2^2$. This objective groups similar contextual motion features into a shared discrete vocabulary.

The resulting representation is a discrete motion-token sequence $\mathcal{Z} = (z_1, \ldots, z_M)$, with each token accompanied by its timestamp on the uniform temporal grid. Token positions are independent of annotated or predicted boundaries; transition supervision shapes the encoded features rather than token locations. Both f_ϕ and $\mathcal{C}$ remain fixed during downstream language adaptation.

3.3 Language Model Integration

To recover subtask boundaries and language descriptions from the discrete motion representation, we adapt a pretrained language model to interpret motion tokens in their temporal and task context. We add $K = 512$ motion entries, `<k0>` through `<k511>`, to the language-model vocabulary, each with a trainable embedding. The input combines a task instruction with the motion-token sequence $\mathcal{Z} = (z_1, \ldots, z_M)$ defined above. We insert textual time markers every second to provide a temporal reference for the predicted segment boundaries.

For supervision, we serialize the reference segmentation as a JSON sequence of segments, each containing a start time, an end time, and a natural-language description. Segment times are expressed in seconds from the start of the trajectory. Let $Y = (y_1, \ldots, y_L)$ denote the tokenized JSON response, where L is its length and ℓ indexes language-model output tokens rather than motion time steps. Given the input prompt P, containing the task instruction, motion tokens, and time markers, we optimize $\mathcal{L}_{\text{read}} = -\sum_{\ell=1}^{L} \log p_\theta(y_\ell \mid y_{<\ell}, P)$, where θ denotes the adapted reader parameters and $y_{<\ell}$ denotes the preceding response tokens. The loss is applied only to response tokens.

Table 1: **Training and evaluation splits.** Duration and segments per episode are reported as mean ± SD. The final column reports the number of distinct action verbs, with the percentage of segments whose verb occurs in the ABC-130K training annotations in parentheses.

Dataset	Embodiment	Purpose	Episodes	Tasks	Duration (s)	Segments per ep.	Seg. length (s)	Bnd. per min	Shared Verbs (in train)
Training									
ABC-130K	YAM	Training	5,478	140	173.3 ± 66.5	26.6 ± 15.0	4.1	9.2	83 (100%)
Evaluation									
ABC-130K	YAM	Held-out Test	200	10	154.6 ± 57.6	24.3 ± 13.4	3.9	9.4	24 (99%)
AgiBot World	AgiBot G1	Cross-Robot-Emb.	2,653	3	94.4 ± 31.1	10.8 ± 3.1	6.8	6.9	12 (61%)
Ego-Exo4D	Human hands	Transfer to Human	193	12	81.4 ± 53.5	10.1 ± 5.9	4.8	7.4	69 (35%)

4 EXPERIMENTS

We evaluate KiRoSe across trajectory segmentation, cross-embodiment transfer, language generalization, and task-progress understanding. The experiments address four research questions: (1) Can KiRoSe identify semantic subtask boundaries and descriptions on unseen manipulation trajectories? (2) Can a motion representation trained on one robot transfer to unseen embodiments without additional training? (3) Does KiRoSe transfer action semantics to new embodiments, and does adaptation preserve the language model's vocabulary beyond the training annotations? (4) Does the learned motion representation encode information about task progress, including whether an action is being executed forward or reversed?

4.1 IMPLEMENTATION AND PROTOCOL

Bimanual-Arm Representation. To support transfer across robots and human hands, we use a shared end-effector motion representation:

$$\mathbf{x}_t = \bigoplus_{h \in L,R} [\mathbf{v}_{h,t}^\top, \boldsymbol{\omega}_{h,t}^\top, \dot{g}_{h,t}]^\top \in \mathbb{R}^{14}, \tag{1}$$

where $\bigoplus$ concatenates left/right linear velocity ($\mathrm{m/s}$), angular velocity ($\mathrm{rad/s}$), and normalized opening rate (s^{-1}). For robots, these signals come from end-effector motion and gripper opening; for humans, from wrist motion, palm orientation, and thumb–index separation.

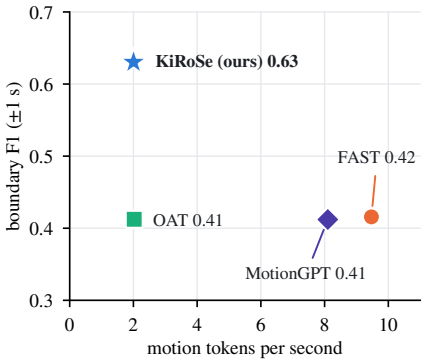
Datasets and Evaluation Splits. Table 1 summarizes the training and evaluation splits. Training uses 5,478 episodes from the YAM bimanual robot in ABC-130K (XDOF, 2026), spanning 140 manipulation tasks. Evaluation is conducted on three splits. The first contains 200 held-out ABC-130K episodes from the same YAM embodiment. Cross-embodiment evaluation uses 2,653 episodes in AgiBot World (Bu et al., 2025) and 193 human-hand trajectories in Ego-Exo4D (Grauman et al., 2024). The AgiBot and Ego-Exo4D tasks are not present in the training set. Some baselines are evaluated on subsets for the cost reason; each table and figure states its episodes.

These splits differ in embodiment, motion characteristics, and temporal structure (Figure 3). AgiBot and Ego-Exo4D contain fewer segments and annotated boundaries per minute than ABC-130K. Semantic overlap also decreases: training verbs cover 99% of held-out ABC-130K segments, compared with 61% for AgiBot and 35% for Ego-Exo4D (Table 1). Evaluation thus tests generalization across episodes, embodiments, task semantics, and temporal structure.

Metrics. We use complementary metrics to evaluate the temporal and semantic information of robot demonstration semantic segmentation, corresponding to two questions: *when* manipulation actions change and *what* action occurs within each segment. *Boundary F1*, a metric introduced by (Chen et al., 2026a) to measure temporal segmentation accuracy independently of the predicted action description by matching predicted and reference segment boundaries one-to-one within a temporal tolerance and macro-averaging F1 across episodes; multiple tolerances are reported to distinguish precise boundary localization from approximate transition detection. *Grounded F1* evaluates temporal and semantic correctness jointly: predicted and reference segments are first matched one-to-one when their interval IoU is at least 0.5, and a match is counted as correct only when the corresponding

Table 2: **Demonstration semantic segmentation on 200 YAM episodes from 10 held-out tasks. Left:** Boundary F1@$\pm$0.5/1/2 s and Grounded F1 using lexical or embedding-based matching. FAST, OAT, MotionGPT, and KiRoSe use the same LoRA-adapted Qwen3.5-9B reader; baseline input and training-set differences are listed in Appendix C. †Video and task instruction. **Right:** Boundary F1@$\pm$1 s as function of motion-token rate.

	Boundary F1			Grounded F1	
Method	$\pm$0.5 s	$\pm$1 s	$\pm$2 s	Lex.	Emb.
Qwen3.5-9B zero-shot†	0.09	0.16	0.22	0.05	0.07
FAST	0.22	0.42	0.50	0.12	0.17
OAT	0.22	0.41	0.51	0.14	0.20
MotionGPT	0.23	0.41	0.50	0.13	0.16
KiRoSe: Latent + MLP	0.44	0.52	0.56	–	–
KiRoSe: SMT codes + MLP	0.38	0.45	0.49	–	–
KiRoSe	**0.48**	**0.63**	**0.71**	**0.43**	**0.49**



action descriptions also agree. We report two variants: Lexical Grounded-F1 requires normalized description-token F1 ≥ 0.5, while Embedding Grounded-F1 requires MiniLM cosine similarity ≥ 0.6. The lexical metric emphasizes agreement in action vocabulary, whereas the embedding metric allows semantically equivalent descriptions with different wording. Control annotations such as `mistake`, `pause`, `wait`, `idle`, and `reset` are excluded from action-description scoring.

4.2 Trajectory Segmentation

We compare KiRoSe with FAST (Pertsch et al., 2025), OAT (Liu et al., 2026), and MotionGPT (Jiang et al., 2023), pairing each tokenizer with a Qwen3.5-9B (Qwen Team, 2026) reader adapted using the same LoRA training procedure. To examine the contributions of the learned features and language-model reader, we also evaluate two variants of our method: *Latent + MLP*, which predicts boundaries directly from continuous encoder features, and *SMT codes + MLP*, which uses the quantized representation but replacing language model with a multilayer perception head. All methods are evaluated on 200 unseen YAM episodes from 10 held-out tasks.

We first evaluate whether the motion representation preserves signals that are informative for semantic action transitions. Even reconstruction-based kinematic tokenizers such as FAST and OAT improve Boundary F1 over the zero-shot Qwen3.5-9B video baseline, suggesting that kinematics provide a more direct signal for localizing manipulation transitions than visual observations alone in this setting. SMT further improves over FAST, OAT, and MotionGPT across all temporal tolerances (Table 2). Because these tokenizers are paired with the same Qwen3.5-9B reader and adaptation procedure, the remaining difference is primarily associated with how the motion signal is represented. Reconstruction-oriented tokenizers are designed to preserve low-level trajectory detail, whereas KiRoSe is trained to emphasize changes in motion that coincide with subtask transitions. Signals such as changes in end-effector velocity, rotation, and gripper state are therefore represented in a form more directly aligned with manipulation boundaries, yielding 0.63 Boundary F1 at $\pm$1 s compared with approximately 0.41–0.42 for the alternative kinematic tokenizers.

The evaluation data also suggest that the motion representation retains semantic information beyond transition timing. KiRoSe obtains higher lexical and embedding Grounded F1 than the alternative kinematic tokenizers, indicating that predicted segments are more often paired with descriptions consistent with the corresponding reference actions. Therefore, SMT-based transition-supervised motion tokens preserve information about both when manipulation actions change and the motion context needed to interpret what action is being performed.

4.3 Cross Embodiment Generalization

We evaluate KiRoSe in its generalization capability for demonstrating semantic segmentation along two dimensions: transfer of transition timing and transfer of action semantics. As summarized in Table 1, all trainable components are learned exclusively from ABC-130K trajectories collected

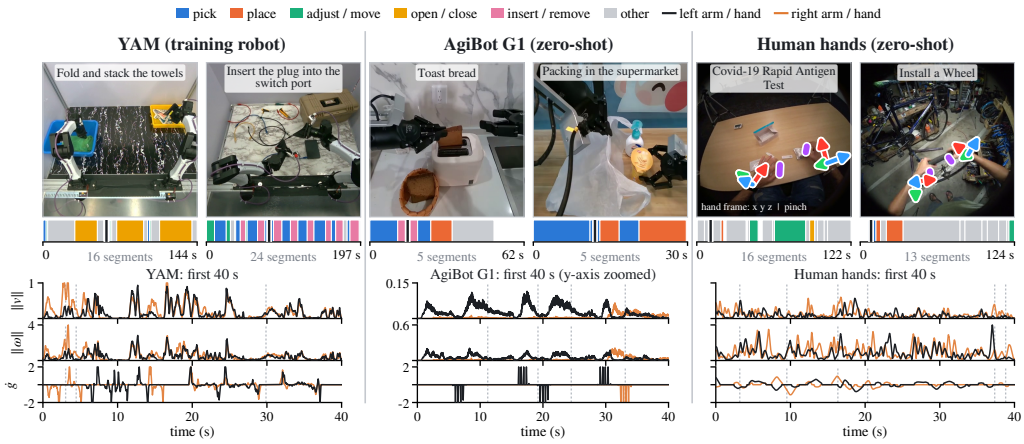


Figure 3: **Evaluation Dataset.** Top: example tasks and annotated action timelines, with colors indicating action categories. Bottom: representative 40-second traces of linear speed, angular speed, and opening rate for the left (black) and right (orange) arms or hands. The datasets differ in task semantics, temporal structure, and motion characteristics.

with YAM, while AgiBot G1 and human-hand trajectories are evaluated without target-embodiment training. Figure 3 illustrates the resulting domain shift: the three embodiments share the same end-effector representation but differ in motion statistics, temporal structure, visual appearance, and annotation semantics.

Transfer of Transition Timing. We first examine whether the temporal structure learned from YAM transfers to unseen embodiments. Figure 4 reports Boundary F1 to measure *when* the kinematic event happens.

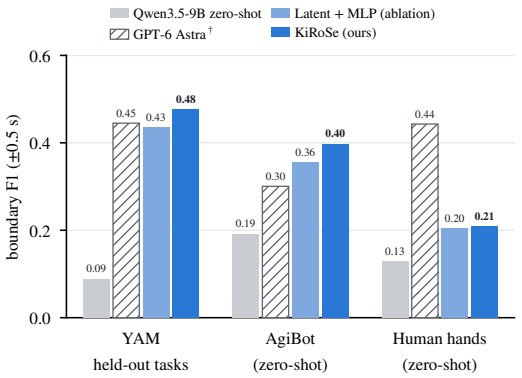


Figure 4: Boundary F1 (±0.5 s) per embodiment. KiRoSe is trained on YAM only; AgiBot and human hands are zero-shot.

The transfer from YAM to AgiBot involves substantial domain shifts: the two datasets use different robot embodiments and teleoperation interfaces, exhibit different motion statistics, and differ in both segment duration and annotation granularity (Figure 3 and Table 1). Despite these differences, KiRoSe transfers without any AgiBot training data, obtaining 0.40 Boundary F1 at ±0.5 s AgiBot testset, compared with 0.48 on held-out YAM trajectories. We ablate with latent + MLP variant, which predicts boundaries directly from the learned motion features, also transfers effectively, achieving 0.36 on AgiBot. The video-based Qwen3.5-9B baseline obtains 0.19 on AgiBot episodes with video. We also measure the performance of state-of-the-art vision-language model GPT-6 Astra and the results suggest that KiRoSe outperforms Astra for hold-out set performance on YAM and AgiBot zero-shot transfer. The retained boundary localization across these domain shifts suggests that transition-supervised motion features capture changes in end-effector motion that are shared across robot embodiments, rather than transition patterns specific to YAM. Removing the motion tokens or replacing them with tokens from another trajectory reduces Boundary F1 on the two robot embodiments, further indicating that the transferred predictions depend on the trajectory-specific motion signal.

Action Semantic Generalization. We next evaluate action semantics on the AgiBot episodes by comparing with GPT-6 Astra, using two Grounded F1 variants. *Action-level* Grounded F1 requires only the verb class to match; *full-description* Grounded F1 also requires the wording, including object names, to match. KiRoSe reaches 0.59 action-level Grounded F1 versus 0.34 for Astra.

Table 3: **Unseen-task vocabulary on Ego-Exo4D.** Supplied words appear in the task's annotations but not in YAM training or the prompt. Hits: trajectories with a supplied word.

Task	Traj.	Hits	Supplied words
Omelet	36	17	pan, egg, mixture, whisk
COVID-19 test	65	14	swab
Tomato & eggs	12	11	pan
Flat tire	13	8	inner (tube), lever
Noodles	9	3	pot, stir
Salad	26	2	toss, dressing

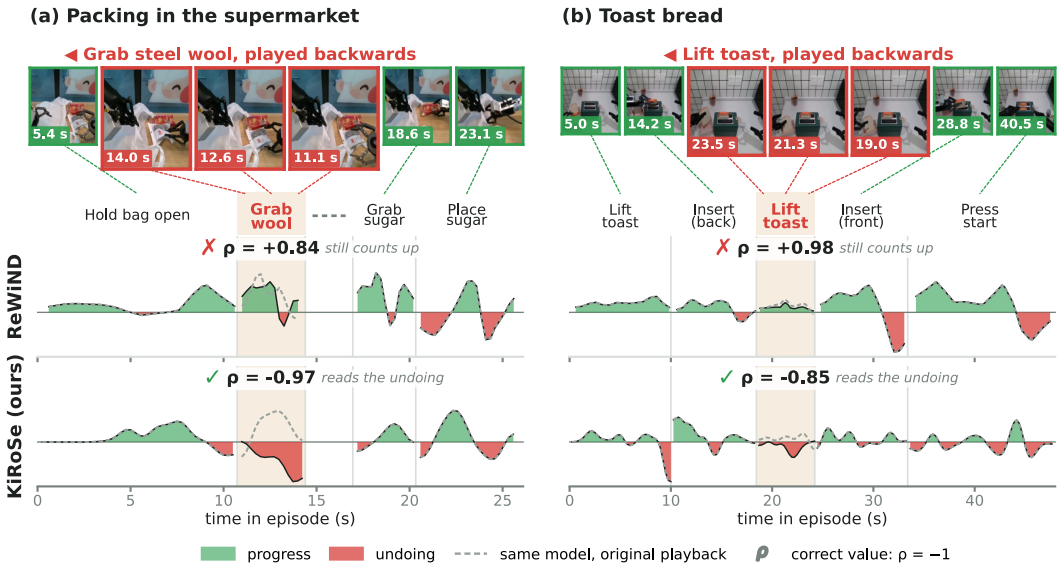


Figure 5: **Qualitative reversal probe on AgiBot (zero-shot).** One subtask per episode is played backwards in place (red frames; stamps give original recording time). Curves show the signed progress rate (green: rising, red: falling); dashed: same model on the original video. ReWiND keeps counting up while the task is undone; KiRoSe's progress falls; Table 4 reports statistic result.

Human activities in Ego-Exo4D introduce a larger semantic shift: only 35% of annotated segments use verbs observed during training, compared with 61% for AgiBot (Table 1). We therefore complement the robot evaluation with a qualitative analysis on Ego-Exo4D human-hand trajectories from 12 tasks that are absent from YAM training. YAM-trained KeRoSe receives only the task name and motion-token sequence. As reported in Table 3, supplied words occur in 6 of the 12 tasks, including "tube" and "lever" for Flat tire installation, "swab" for a COVID-19 test, and "whisk" and "egg" for preparing an omelet. These words are consistent with what a language model can infer from the task name, so this result suggests that as LoRA adaptation with KeRoSe preserves the language model's open-vocabulary knowledge, rather than as semantics recovered from the training dataset .

4.4 Cross-Embodiment Progress Estimation

We further hypothesize that transition-supervised motion representations also retain ordered motion patterns *within* a segment. For example, this may indicate whether an action is progressing toward completion or being undone. A useful dense progress signal should increase during successful execution and decrease when an action regresses: ReWiND (Zhang et al., 2025), for example, uses reversed trajectories to represent decreasing progress. We therefore use action reversal as a controlled test: if a model represents progress rather than elapsed time, reversing an action should reverse its predicted progress.

Table 4: **Zero-shot reversal progress evaluation on AgiBot.** The same 2,586 subtasks (357 episodes) are evaluated forward and reversed. "Rev. $\rho < 0$" is the percentage of reversed subtasks with negative progress correlation.

Model	Fwd. $\rho \uparrow$	Rev. $\rho \downarrow$	Rev. $\rho < 0$ (%) $\uparrow$
ReWiND (MW)	$+\mathbf{0.88}$	$+0.88$	1%
ReWiND (OXE)	$+0.65$	$+0.63$	4%
Robometer-4B	$+0.71$	$+0.06$	46%
KiRoSe	$+0.51$	$-\mathbf{0.68}$	**86%**

Setup. We use 2,586 annotated subtasks of at least 3.2 s from 357 AgiBot episodes across three tasks. Each subtask is evaluated as recorded and reversed in place, with the rest of the episode unchanged: video frames are played backward and kinematic velocities are negated, which corresponds to motion that undoes the action. We compare with ReWiND (Zhang et al., 2025), retrained with the authors' public code on MetaWorld (MW) and on Open-X (OXE), and with the released Robometer-4B (Liang et al., 2026); both are trained with supervision that includes reversed or rewound examples. For KiRoSe, the 2.76M-parameter SMT encoder, trained only for transition detection on YAM, stays frozen. A 68k-parameter probe reads its features and the displacement accumulated since the subtask start, and is trained only on YAM subtasks (Appendix E). We report the Spearman correlation ρ between predicted progress and playback time ($\rho > 0$ forward, $\rho < 0$ when undone), and the fraction of reversed subtasks with $\rho < 0$. A model that tracks elapsed time stays positive in both directions.

Results and Analysis. Table 4 compares whether predicted progress follows the direction of action execution under reversal. Overall, SMT maintains a positive progress correlation during forward execution and switches to a negative correlation when the same actions are reversed. In contrast, the video-based reward models remain positively correlated or approach zero under reversal, despite being trained with progress-related supervision that includes reversed or rewound trajectories. On AgiBot, SMT obtains $\rho = +0.51$ on forward trajectories and $\rho = -0.68$ on reversed trajectories, with predicted progress falling on 86% of reversed subtasks. ReWiND remains positively correlated after reversal, with $\rho = +0.88$ and $+0.63$ for its MetaWorld and Open-X checkpoints, respectively, while Robometer-4B approaches zero at $\rho = +0.06$. Because neither the SMT-encoder nor the lightweight head sees AgiBot, this suggests that transition-supervised kinematic features retain the direction of within-segment motion in a form that transfers across embodiments. Reversing an action changes the direction of end-effector velocity and reverses motion events such as gripper closing and opening, which are also informative for identifying semantic transitions.

5 Conclusion

We presented KiRoSe, a framework for identifying *when* manipulation actions change and *what* action occurs within each segment. KiRoSe uses SMTs to compress high-frequency kinematics into a transition-supervised representation for language-model interpretation. Evaluation on YAM, AgiBot G1, and Ego-Exo4D indicates that SMT preserves both temporal and semantic information, transfers transition structure zero-shot across robot embodiments together with action identity, and preserves the language model's open-vocabulary knowledge. Beyond segmentation, the same frozen representation supports within-segment progress estimation on an unseen robot, including detecting when an action is being undone, even though the encoder is never trained for progress or reversal. Future work can extend this representation toward richer contact and object-centric signals, broader robot embodiments, and downstream demonstration retrieval, data curation, and reward modeling.

AI Use Statement

We used generative AI tools to assist in revising the kinematic encoder design, including discussing architectural choices and elaborating implementation details. We also used these tools to refine manuscript wording, organization, and figure captions, and to assist with literature searches and LaTeX formatting. All AI-assisted suggestions and content were reviewed by the authors, who take full responsibility for the methodology, experiments, results, and final manuscript.

REFERENCES

Matthew M. Botvinick, Yael Niv, and Andrew C. Barto. Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, 113(3):262–280, 2009.

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Xindong He, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025.

Annie S. Chen, Alec M. Lessing, Yuejiang Liu, and Chelsea Finn. Curating demonstrations using online experience. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.071. URL https://www.roboticsproceedings.org/rss21/p071.html.

Kaiyuan Chen, Shuangyu Xie, Andrew Goldberg, and Ken Goldberg. Learning multi-task robot trajectory segmentation from visual and kinematic streams. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 1452–1461, June 2026a.

Shirui Chen, Cole Harrison, Ying-Chun Lee, Angela Jin Yang, Zhongzheng Ren, Lillian J. Ratliff, Jiafei Duan, Dieter Fox, and Ranjay Krishna. TOPReward: Token probabilities as hidden zero-shot rewards for robotics. *arXiv preprint arXiv:2602.19313*, 2026b. URL https://arxiv.org/abs/2602.19313.

Caelan Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. SkillMimicGen: Automated demonstration generation for efficient skill learning and deployment. In *8th Annual Conference on Robot Learning*, 2024. URL https://arxiv.org/abs/2410.18907.

Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19383–19400, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. URL https://arxiv.org/abs/2106.09685.

Tariq Iqbal, Shen Li, Christopher Fourie, Bradley Hayes, and Julie A Shah. Fast online segmentation of activities from partial trajectories. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5019–5025. IEEE, 2019.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. MotionGPT: Human motion as a foreign language. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.14795.

Sanjay Krishnan, Animesh Garg, Sachin Patil, Colin Lea, Gregory Hager, Pieter Abbeel, and Ken Goldberg. Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning. *The International journal of robotics research*, 36(13-14):1595–1618, 2017.

Sanjay Krishnan, Animesh Garg, Richard Liaw, Brijen Thananjeyan, Lauren Miller, Florian T Pokorny, and Ken Goldberg. Swirl: A sequential windowed inverse reinforcement learning algorithm for robot tasks with delayed rewards. *The international journal of robotics research*, 38 (2-3):126–145, 2019.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.

Seungjae Lee, Yibin Wang, Haritheja Etukuru, H. Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 26991–27008, 2024. URL https://arxiv.org/abs/2403.03181.

Anthony Liang, Yigit Korkmaz, Jiahui Zhang, Minyoung Hwang, Abrar Anwar, Sidhant Kaushik, Aditya Shah, Alex S. Huang, Luke Zettlemoyer, Dieter Fox, Yu Xiang, Anqi Li, Andreea Bobu, Abhishek Gupta, Stephen Tu, Erdem Biyik, and Jesse Zhang. Robometer: Scaling general-purpose robotic reward models via trajectory comparisons. In *Robotics: Science and Systems (RSS)*, 2026. URL https://arxiv.org/abs/2603.02115.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

Chaoqi Liu, Xiaoshen Han, Jiawei Gao, Yue Zhao, Haonan Chen, and Yilun Du. OAT: Ordered action tokenization. *arXiv preprint arXiv:2602.04215*, 2026. URL https://arxiv.org/abs/2602.04215.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, 2023. URL https://arxiv.org/abs/2310.17596.

Marius Memmel, Jacob Berg, Bingqing Chen, Abhishek Gupta, and Jonathan Francis. STRAP: Robot sub-trajectory retrieval for augmented policy learning. *arXiv preprint arXiv:2412.15182*, 2024. URL https://arxiv.org/abs/2412.15182.

Pietro Morasso. Three dimensional arm trajectories. *Biological Cybernetics*, 48(3):187–194, 1983.

Open X-Embodiment Collaboration. Open X-embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2023. URL https://arxiv.org/abs/2310.08864.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2304.07193.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. URL https://arxiv.org/abs/2501.09747.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Dagmar Sternad and Stefan Schaal. Segmentation of endpoint trajectories does not imply segmented control. *Experimental Brain Research*, 124(1):118–136, 1999.

Daniel Tanneberg, Kai Ploeger, Elmar Rueckert, and Jan Peters. Skid raw: Skill discovery from raw trajectories. *IEEE robotics and automation letters*, 6(3):4696–4703, 2021.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Paolo Viviani and Massimo Cenzato. Segmentation and coupling in complex movements. *Journal of Experimental Psychology: Human Perception and Performance*, 11(6):828–839, 1985.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Bizhu Wu, Jinheng Xie, Keming Shen, Zhe Kong, Jianfeng Ren, Ruibin Bai, Rong Qu, and Linlin Shen. MG-MotionLLM: A unified framework for motion comprehension and generation across multiple granularities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27849–27858, 2025. URL https://arxiv.org/abs/2504.02478.

XDOF. ABC-130k. Dataset release, 2026. URL https://huggingface.co/datasets/XDOF/ABC-130k.

Jiahui Zhang, Yusen Luo, Abrar Anwar, Sumedh Anand Sontakke, Joseph J Lim, Jesse Thomason, Erdem Biyik, and Jesse Zhang. ReWiND: Language-guided rewards teach robot policies without new demonstrations. In *Conference on Robot Learning (CoRL)*, 2025. URL https://arxiv.org/abs/2505.10911. Oral.

Zixiang Zhou, Yu Wan, and Baoyuan Wang. AvatarGPT: All-in-one framework for motion understanding, planning, generation and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2311.16468.