RONDO: UNSUPERVISED DISCOVERY OF RECURRING TEMPORAL STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Many real-world time series data exhibit structural properties at multiple scales, from short, recurring units to complex sequences composed of these units. Unsupervised discovery of both these components and structure enables the design of intelligent systems that help interpreting temporal data thereby limiting the amount of costly human annotations required. Existing modeling approaches typically overlook the hierarchical structure inherent to many time series, treating recurring patterns at different temporal scales as independent structures. Moreover, most assume access to the complete data sequence and treat discovery as a static process, limiting their ability to evolve as the stream grows.

We introduce RONDO, an unsupervised approach for modeling recurring hierarchical structure in continuous temporal streams. By explicitly constructing vocabularies of reusable units and their recurring compositions, RONDO captures structure shared across complex temporal patterns while refining and expanding its discoveries as the stream evolves. Evaluations on temporal sequences spanning diverse domains and data modalities show that RONDO outperforms existing unsupervised recurrence-discovery baselines, with particularly pronounced advantages in limited-data and continual-stream settings. These capabilities support recurring-pattern discovery, scalable behavior understanding, and adaptive intelligent systems operating on long, unlabeled temporal streams.

1 INTRODUCTION

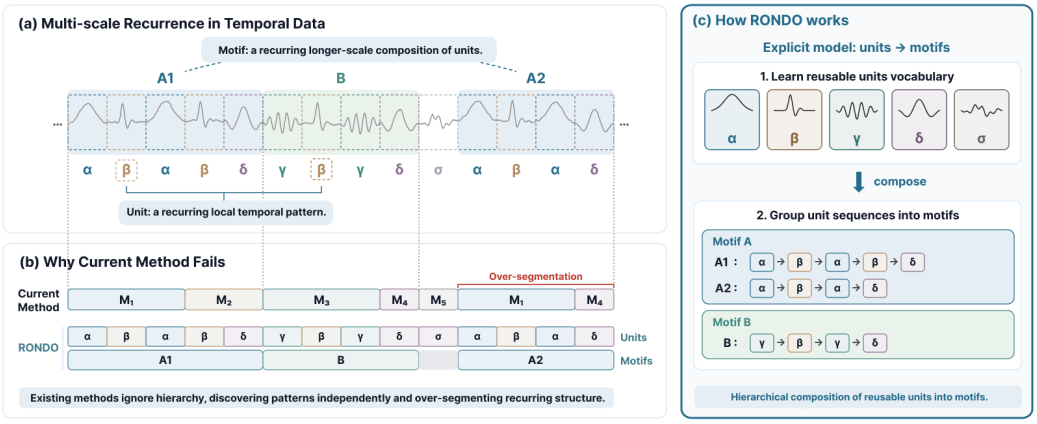


Figure 1: RONDO discovers recurring temporal structure hierarchically. (a) Temporal streams exhibit recurring local units that compose into longer-scale motifs. (b) Current methods can fragment such structure and cause over-segmentation, while RONDO preserves the unit-motif hierarchy. (c) RONDO learns reusable units and composes recurring unit sequences into motifs.

Continuous temporal data often exhibit recurring structure across multiple temporal scales. At a local scale, subsequences that differ in their signals, duration, or surrounding context may still rep-

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.

resent the same underlying temporal unit (Zhang et al., 2012). These recurring patterns form a compact vocabulary of reusable **units**. At a longer time scale, units are organized into recurring **motifs**. Different instances of the same motif may contain repeated or missing units, or have slightly different boundaries, while still preserving a similar overall organization (Kuehne et al., 2014). This phenomenon appears across diverse domains, including human motion, driving, and logistics. For example, daily activities such as food preparation may reuse local motion units such as reaching for objects or changing posture, while differing in how these units are ordered and combined into longer motifs (Kipf et al., 2019; Niemann et al., 2020; Martin et al., 2019; Chavarriaga et al., 2013; Maes et al., 2021). We refer to this nested organization as **multiscale compositional recurrence**, illustrated in Fig. 1(a). Explicitly separating these levels allows shared local structure to be reused across multiple higher-level patterns rather than treating each realization as an independent discovery: accumulated data refine the unit vocabulary, while novel arrangements expand the motif vocabulary.

Existing unsupervised methods capture individual aspects of this structure but rarely model the interaction between temporal scales. Early clustering and discovery approaches focus on recurring local states within a sequence (Matsubara et al., 2014; Hallac et al., 2017). Motif-discovery methods extend this perspective to repeated subsequences over longer horizons (Minnen et al., 2006), while more recent representation-learning and hierarchical segmentation approaches capture temporal structure at multiple resolutions or recover reusable components (Yue et al., 2022; Liu & Chen, 2024; Kipf et al., 2019; Wu et al., 2022). However, these levels are typically discovered or represented in isolation, leaving the connections between local structure and longer recurring patterns largely implicit. As illustrated in Fig. 1(b), this can fragment related temporal patterns into separate discoveries when the same local structures occur with different repetitions and arrangements.

In this work, we explicitly model the hierarchical organization of temporal streams. We introduce RONDO, an approach that discovers and maintains complementary vocabularies at different scales (Fig. 1(c)). At the local level, **UnitAlign** adapts generic temporal representations to learn a vocabulary of recurring and reusable units, while **MotifFormer** models their temporal organization to discover higher-level motifs. This separation enables local units to be reused across different motifs while preserving their distinct compositions. As new data arrive, UnitAlign first updates the unit vocabulary by refining matched units or introducing recurring novel ones; MotifFormer then operates on the updated unit sequence to refine existing motifs or add new compositions.

We evaluate RONDO across diverse temporal datasets against unsupervised clustering, segmentation, and recurring-pattern discovery approaches. Across datasets, RONDO improves the recovery of recurring structure, with particularly strong gains when data are initially limited and subsequently expanded. Ablation studies show that both unit-level adaptation and higher-level motif modeling contribute to these improvements, while experiments with different temporal representations demonstrate that the gains are not tied to a particular representation backbone.

Our contributions are:

- We formulate unsupervised recurring-structure discovery as hierarchical temporal modeling across multiple scales with explicitly reusable local structure.
- We introduce RONDO, combining **UnitAlign** for reusable unit discovery and **MotifFormer** for modeling their higher-level recurring compositions.
- We demonstrate consistent improvements over existing unsupervised baselines across diverse temporal datasets, sensing modalities, and sequence characteristics.
- We show that learned unit vocabularies can transfer across records to support motif discovery in new streams without relearning local structure.
- We enable continual refinement and expansion of unit and motif vocabularies as new recurring structure emerges over evolving temporal streams.

2 RELATED WORK

2.1 UNSUPERVISED TEMPORAL STRUCTURE DISCOVERY

Recurring structure in temporal data has traditionally been studied through motif discovery. Classical methods identify approximately repeated subsequences and extend this formulation to multi-

dimensional, variable-length, and streaming settings (Chiu et al., 2003; Mueen et al., 2009; Mueen & Keogh, 2010; Yeh et al., 2016; Linardi et al., 2018). These methods locate repeated temporal patterns directly from raw data without predefined semantic categories or boundaries, but generally treat motifs as independent subsequences. As a result, related structure that varies in duration, context, or composition may be fragmented across multiple discovered patterns.

More recent approaches incorporate richer temporal context when modeling long sequences. TICC, Time2State, and E2USD infer recurring states or regimes from multivariate streams, with E2USD further supporting efficient online state discovery (Hallac et al., 2017; Wang et al., 2023; Lai et al., 2024). Unsupervised action segmentation combines representation learning, clustering, and temporal constraints to obtain coherent segment sequences (Kumar et al., 2022; Xu & Gould, 2024), while TNC, TS2Vec, and TimesURL learn contextual representations over different temporal supports (Tonekaboni et al., 2021; Yue et al., 2022; Liu & Chen, 2024). Despite these advances, recurring structure is still typically represented as a flat set of states or segments, obscuring reuse across temporal scales. We instead explicitly model hierarchical relationships between recurring structures, allowing shared local patterns to be reused within longer temporal organizations.

2.2 Hierarchical and Compositional Temporal Modeling

Hierarchical representations have been widely used to model sequential data at multiple temporal scales. Rather than treating a long sequence as a collection of independent states, these approaches represent longer temporal patterns as compositions of shorter, reusable structures (Sutton et al., 1999; Niekum et al., 2012; Kipf et al., 2019; Jiang et al., 2022). Such representations allow local structure to be shared across different temporal contexts and provide a natural way to capture compositional organization in long sequences.

This hierarchical view has also been explored in unsupervised temporal discovery. Existing approaches construct higher-level structure by recursively grouping lower-level temporal units, modeling compositional relations among discovered segments, or deriving coarse patterns from finer-grained structure (Wu et al., 2022; Gong et al., 2023; Spurio et al., 2025). Related work in smart-home sensing similarly reuses previously discovered local patterns to construct higher-level activity representations (Hiremath et al., 2022). These studies show that discovered temporal structure need not be represented as a flat set of patterns. However, most existing approaches learn such hierarchies from fixed datasets or within specific domains, with limited support for revising them as new recurring structure emerges. We extend this direction to continual unsupervised discovery, where both local and higher-level structure can evolve with the continual stream.

2.3 Continual and Transferable Structure Discovery

Temporal discovery becomes more challenging when the stream arrives progressively. Classical stream-clustering methods such as CluStream and DenStream maintain and update cluster structure over evolving data streams (Aggarwal et al., 2003; Cao et al., 2006), while online motif discovery and systems such as StreamScope incrementally track recurring temporal patterns (Mueen & Keogh, 2010; Kawabata et al., 2018). Related challenges have been studied in open-world and generalized category discovery, where models distinguish known from unseen categories and, in continual or online settings, incorporate novel categories while preserving previously discovered structure (Vaze et al., 2022; Wu et al., 2023; Park et al., 2024; Ma et al., 2024; Dai & Chauhan, 2025).

Despite their different settings, these approaches generally model novelty at a single level, assigning each segment to an atomic category; recent temporal action-discovery methods similarly use labeled known actions to define the granularity of novel ones (Spurio et al., 2026). This is restrictive for compositional streams, where novelty may arise within known motifs, through new compositions of known units, or as genuinely new units, while local structure may remain reusable even when higher-level compositions change. We therefore maintain separate unit- and motif-level vocabularies, allowing structure to be updated and transferred where recurring regularities persist or emerge.

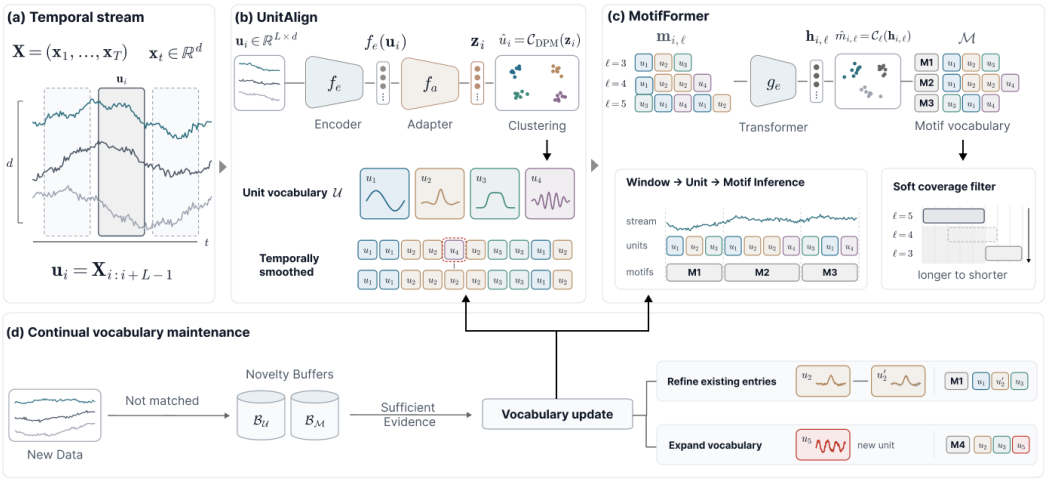


Figure 2: Overview of RONDO. (a) A continuous temporal stream is divided into sliding windows. (b) UnitAlign encodes and adapts each window, clusters recurring local structure into a reusable unit vocabulary, and smooths unit assignments. (c) MotifFormer models unit sequences across temporal scales to discover higher-level motifs, followed by a soft coverage filter. (d) During continual operation, unmatched windows and unit subsequences are stored in novelty buffers and, once sufficient evidence is available, used to refine existing entries or introduce new units and motifs.

3 METHODOLOGY

We now describe **RONDO**, an unsupervised approach for discovering recurring, hierarchical temporal structure from continuous timeseries. As illustrated in Fig. 2, the approach consists of two main stages. **UnitAlign** adapts local temporal representations and organizes recurring patterns into a shared unit vocabulary. **MotifFormer** then models the resulting unit sequence to discover recurring higher-level motifs. As the stream grows, both vocabularies are progressively updated. We first formalize the discovery problem before describing each component in detail.

3.1 PROBLEM FORMULATION

Let a continuous multivariate temporal stream be

$$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T), \qquad \mathbf{x}_t \in \mathbb{R}^d, \tag{1}$$

where T is the stream length and d the number of channels. The general objective of unsupervised recurring pattern discovery is to identify temporal patterns that repeatedly occur in $\mathbf{X}$ without known temporal boundaries or semantic labels. We define each motif instance and its latent type as

$$\mathbf{m}_i = \mathbf{X}_{s_i:e_i}, \qquad \hat{m}_i \in \mathcal{M} = \{1, \ldots, K_m\}, \tag{2}$$

where s_i and e_i denote the unknown temporal boundaries of instance $\mathbf{m}_i$, and $\hat{m}_i$ denotes its motif class. Instances assigned to the same $\hat{m}$ share similar temporal structure despite variations in duration or boundaries. The task is to discover recurring patterns and their occurrences without knowing their number or temporal extent in advance. Temporal recurrence may occur across multiple scales, where shorter reusable structures are repeatedly composed into longer patterns of variable extent (Sanzari et al., 2019; Minnen et al., 2007; Li et al., 2012). We refer to this nested structure as **multi-scale compositional recurrence**. To model this structure, we formulate recurring pattern discovery at two levels: reusable local **units** and higher-level **motifs**. As illustrated in Fig. 2(a), we partition $\mathbf{X}$ into sliding windows $\mathbf{u}_i$, which provide the basic inputs for discovering reusable units.

$$\mathbf{u}_i = \mathbf{X}_{i:i+L-1}, \qquad \hat{u}_i = f(\mathbf{u}_i), \quad \hat{u}_i \in \mathcal{U} = \{1, \ldots, K_u\}., \tag{3}$$

where L is the local window length, $f(\cdot)$ maps each window to a unit class and $\mathcal{U}$ is the discovered unit vocabulary. Applying Eq. 3 across the stream produces the ordered unit sequence, where recurring windows share entries in the unit vocabulary $\mathcal{U}$. At the higher level, each motif ($\hat{m}$) is represented by variable-length compositions of these units, The recurring motif types $\hat{m}$ defined above are then represented at the motif level as variable-length compositions over $\mathcal{U}$. A motif instance is

$$\mathbf{m}_{i,l} \mapsto (\hat{u}_i, \ldots, \hat{u}_{i+\ell-1}), \qquad \hat{m}_{i,\ell} = g(\mathbf{m}_{i,\ell}), \qquad \hat{m}_{i,\ell} \in \mathcal{M} \tag{4}$$

where ℓ is the number of constituent units, $g(\cdot)$ maps each unit sequence to a motif.

3.2 Discovering Reusable Temporal Units

We first instantiate the unit mapping $f(\cdot)$ in Eq. 3 using **UnitAlign**, illustrated in Fig. 2(b). Given each temporal window $\mathbf{u}_i$ from Fig. 2(a), a pretrained self-supervised encoder f_e extracts an initial representation, which an adaptor f_a maps into the latent embedding $\mathbf{z}_i$. DeepDPM then clusters these embeddings and assigns each window to a unit:

$$\mathbf{z}_i = f_a(f_e(\mathbf{u}_i)), \qquad \hat{u}_i = \mathcal{C}_{\mathrm{DPM}}(\mathbf{z}_i), \quad \hat{u}_i \in \mathcal{U}, \tag{5}$$

where $\mathbf{z}_i$ is the adapted representation and $\mathcal{C}_{\mathrm{DPM}}$ denotes the DeepDPM cluster assignment. Each resulting cluster corresponds to one entry in the unit vocabulary $\mathcal{U}$ shown in Fig. 2(b).

The adaptor is optimized with two complementary objectives. A consistency objective preserves the similarity structure of the original SSL representations, while DeepDPM provides pseudo-labels that encourage separability in the adapted space. We alternate between representation adaptation and clustering until the assignments stabilize. Finally, we temporally smooth the resulting assignments to suppress isolated transitions, producing the stable unit sequence at the bottom of Fig. 2(b), which serves as input to MotifFormer.

3.3 Discovering Compositional Motifs

We then instantiate the motif mapping $g(\cdot)$ in Eq. 4 using **MotifFormer**, illustrated in Fig. 2(c). Starting from the smoothed unit sequence produced by UnitAlign, we construct unit subsequences of different lengths ℓ. A Transformer encoder g_e maps each subsequence $\mathbf{m}_{i,\ell}$ into a motif representation $\mathbf{h}_{i,\ell}$, which is then clustered into a recurring motif:

$$\mathbf{h}_{i,\ell} = g_e(\mathbf{m}_{i,l}) = g_e(\hat{u}_i, \ldots, \hat{u}_{i+\ell-1}), \qquad \hat{m}_{i,\ell} = \mathcal{C}_\ell(\mathbf{h}_{i,\ell}), \quad \hat{m}_{i,\ell} \in \mathcal{M}, \tag{6}$$

where $\mathcal{C}_\ell$ denotes the clustering assignment at temporal scale ℓ. The Transformer captures contextual dependencies among units while mapping variable-length sequences into a shared space.

To make these representations robust across local variation and temporal extent, MotifFormer is optimized with two complementary objectives. A consistency objective encourages representations to remain stable under perturbations of the unit sequence and aligns subsequences anchored at the same starting point across different lengths, promoting continuity across temporal scales. In parallel, clustering-derived pseudo-labels are used to improve the separability of motif representations.

After training, we cluster subsequences separately at each temporal scale. As shown in the lower-right of Fig. 2(c), a **soft coverage filter** processes candidates from longer to shorter scales, down-weighting regions already explained by high-confidence long motifs. This reduces redundant re-discovery of already captured structure and allows shorter recurring patterns to emerge. Finally, we aggregate evidence across scales based on each subsequence's proximity to its cluster center to determine the final motif boundaries and assignments.

3.4 Continual Vocabulary Maintenance

Finally, Fig. 2(d) shows how both vocabularies evolve as the stream evolves. When a new window or unit subsequence cannot be confidently matched to an existing entry, it is stored in separate unit and motif novelty buffers, $\mathcal{B}_U$ and $\mathcal{B}_M$, rather than immediately creating a new vocabulary item.

We manage this evolution through dynamically maintained micro-clusters that summarize and adapt the current vocabulary structure. Once sufficient evidence accumulates, buffered windows and subsequences are integrated with the existing micro-clusters to trigger a vocabulary update. The update either *refines* an existing entry when the buffered data remain compatible with known structure, or *expands* the vocabulary when a persistent new unit or motif emerges.

4 EXPERIMENTS

Our experiments address three questions: (1) how accurately can RONDO recover recurring temporal structure compared with existing unsupervised methods; (2) how effectively can it adapt as new data emerge; and (3) which components and design choices contribute most to its performance.

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate RONDO on six multivariate temporal datasets: **MoCap**, **ActRecTut**, **PAMAP2**, **Drive&Act**, **OPPORTUNITY**, and **LARa** (Zhou et al., 2012; Bulling et al., 2014; Reiss & Stricker, 2012; Martin et al., 2019; Chavarriaga et al., 2013; Niemann et al., 2020; Rueda & Fink, 2021). These datasets span diverse sensing environments, class vocabularies, and sequence characteristics, allowing us to evaluate recurring structure discovery across heterogeneous continuous streams. We treat individual recordings as independent sequences and apply dataset-specific preprocessing and window configurations summarized in Appendix C.

Baselines. We compare RONDO against unsupervised temporal state and structure discovery methods, including **AutoPlait** (Matsubara et al., 2014), **TICC** (Hallac et al., 2017), **Time2State** (Wang et al., 2023), **E2USD** (Lai et al., 2024), **CompILE** (Kipf et al., 2019), and **CLaP** (Ermshaus et al., 2025). For continual discovery, we additionally compare against the streaming clustering methods **DenStream** (Cao et al., 2006) and **DBSTREAM** (Hahsler & Bolaños, 2016). Together, these baselines cover regime segmentation, representation-based state discovery, temporal discovery, and online density-based clustering.

Implementation Details. We use the publicly released implementations of the baselines and follow their original configurations whenever applicable. To ensure a consistent comparison, all methods operate on the same recordings and are evaluated on a common temporal axis. We retain each method's original learning and inference procedure while adapting input preprocessing and dataset-specific configurations where necessary. Detailed preprocessing choices, hyperparameters, and implementation settings are provided in Appendices C and E.

4.2 EVALUATION PROTOCOL AND METRICS

We evaluate all methods on a common temporal axis using a per-recording many-to-one mapping from discovered clusters to ground-truth classes. Each cluster is assigned to the class with the largest duration-weighted temporal overlap, allowing multiple clusters to represent different realizations of the same underlying pattern. This mapping is used only for evaluation and follows prior over-clustering and temporal-discovery protocols (Peng et al., 2020; Chen et al., 2025). The **Other** label is retained as a background class.

We report MoF and discovery recall for overall discovery quality, and Edit and segmental F1 at IoU thresholds of 10%, 25%, and 50% for temporal consistency (Lea et al., 2016; 2017). We additionally report the segment ratio (SegRatio), defined as the number of predicted segments relative to ground-truth segments, to characterize over- and under-segmentation (Krishna et al., 2014; Furnari et al., 2017). For continual discovery, we report H-score and forgetting M_f to measure adaptation to emerging structure and retention of previously discovered patterns (Vaze et al., 2022; Wen et al., 2023; Wu et al., 2023; Park et al., 2024). The main text reports MoF, Recall, Edit, F1@50, and SegRatio; complete results are provided in the Appendix 11.

4.3 RECURRING STRUCTURE DISCOVERY

We first evaluate whether RONDO can recover recurring structure from temporal streams. We compare against six baselines across all six datasets using the evaluation protocol described in Sec-

Table 1: Recurring structure discovery results (%). Stochastic methods report mean performance over five random seeds, with standard deviations provided in Appendix F; deterministic or officially fixed-seed methods are evaluated once. CompILE is evaluated once due to its computational cost. Higher is better for MoF, Recall, F1@50, and Edit. For SegRatio, values closer to 1 are better, so ranked with $|\log \mathrm{SegRatio}|$ and lower is better. Best in **bold**, second-best underlined.

Method	MoF	Recall	F1@50	Edit	SegRatio	Method	MoF	Recall	F1@50	Edit	SegRatio
	MoCap						**Drive&Act**				
Time2State	88.68	91.49	84.48	83.80	1.70	Time2State	54.23	29.55	7.00	12.97	11.31
E2USD	84.97	87.45	77.95	76.48	1.97	E2USD	47.60	18.90	5.66	11.80	14.47
TICC	86.61	84.52	79.39	83.18	2.19	TICC	49.47	23.71	6.61	13.01	9.14
AutoPlait	**90.27**	90.28	91.18	92.05	**0.97**	AutoPlait	39.42	8.08	0.26	4.32	0.04
CompILE	66.45	61.05	52.19	63.14	1.97	CompILE	46.17	17.20	9.04	**32.69**	4.13
CLaP	74.61	65.63	60.53	62.39	0.74	CLaP	49.26	17.74	11.43	20.10	**0.37**
RONDO	88.91	**96.78**	**95.06**	**96.09**	2.42	RONDO	**68.36**	**62.30**	**14.29**	22.22	8.04
	PAMAP2						**OPPORTUNITY**				
Time2State	79.69	92.13	6.17	8.65	24.72	Time2State	69.28	48.94	20.80	**58.15**	2.08
E2USD	76.58	90.21	4.22	7.25	26.96	E2USD	69.02	49.95	26.49	50.32	2.91
TICC	74.71	78.69	15.93	16.42	7.27	TICC	65.91	42.99	13.66	21.28	0.37
AutoPlait	45.22	29.81	21.46	21.45	**0.97**	AutoPlait	37.64	20.83	0.00	0.63	0.01
CompILE	34.11	8.28	4.65	10.41	15.92	CompILE	47.13	25.54	3.44	25.24	0.46
CLaP	77.40	79.70	**50.00**	**51.30**	3.03	CLaP	58.88	29.61	2.82	6.61	0.09
RONDO	**84.12**	**94.25**	22.91	20.82	15.37	RONDO	**79.96**	**59.19**	**30.59**	48.06	**1.23**
	ActRecTut						**LARa**				
Time2State	73.40	49.08	**42.20**	**49.07**	1.10	Time2State	69.91	37.95	27.09	45.89	2.00
E2USD	72.87	46.83	40.41	45.03	**1.01**	E2USD	67.24	35.62	21.81	40.36	2.13
TICC	69.68	41.59	29.33	32.24	0.45	TICC	70.40	35.57	28.34	44.85	1.92
AutoPlait	44.42	14.06	2.04	2.31	0.01	AutoPlait	57.46	17.22	1.36	5.06	0.04
CompILE	45.69	27.96	8.76	31.11	0.44	CompILE	60.03	31.30	11.20	26.13	0.57
CLaP	51.68	27.70	5.45	5.45	0.02	CLaP	65.28	23.93	7.47	10.76	0.16
RONDO	**75.66**	**52.85**	36.71	45.26	0.77	RONDO	**78.44**	**53.01**	**38.50**	**47.34**	**1.31**

tion 4.2. This experiment evaluates both discovery quality and temporal segmentation, capturing whether recurring structure is recovered with coherent organization.

As shown in Table 1, RONDO achieves the highest Recall on all six datasets and the highest MoF on five, slightly trailing only AutoPlait on MoCap, suggesting that it recovers recurring structure more completely and assigns it consistently over time. This advantage holds across datasets with diverse modalities and temporal characteristics. Segment-level results show a more mixed pattern. SegRatio indicates that AutoPlait and CLaP often produce fewer, longer segments, while Time2State and E2USD tend to split the sequence into many short segments. RONDO lies between these two extremes and achieves strong F1 and Edit scores on several datasets. PAMAP2 is an exception, where RONDO recovers the underlying classes well but has a high SegRatio, indicating over-segmentation. Overall, the results suggest that modeling recurring structure at multiple levels improves discovery while maintaining reasonably coherent temporal segmentation.

4.4 ADAPTATION AND TRANSFER ACROSS STREAMS

Continual discovery under progressive streams. We evaluate whether RONDO can preserve and extend previously discovered structure as new observations become available. We focus on **OPPORTUNITY** and **PAMAP2**, which provide long continuous recordings suitable for progressive evaluation. For each dataset, only the first 20% of the stream is used to initialize the unit and motif vocabularies, after which the remaining observations are revealed incrementally. Since representations learned independently from each complete recording, such as TS2Vec embeddings, would expose the model to future observations, we instead use the pretrained MantisV2 encoder in this experiment (Appendix D.1).

We compare against the streaming clustering methods **DenStream** and **DBSTREAM** under the same progressive schedule, together with **Time2State@20** as a static reference trained only on the initial portion of the stream. In addition to standard discovery metrics, we report H-score throughout the stream to characterize continual adaptation and M_f to measure forgetting. As shown in Table 2, RONDO remains competitive in frame-level discovery while producing substantially more coherent

Table 2: Continual discovery on progressively revealed streams. H@40–100 denotes H-score after observing 40–100% of the stream.

Method	MoF	Recall	F1@50	Edit	SegRatio	H@40	H@60	H@80	H@100	M_f
OPPORTUNITY										
Time2State@20	44.70	36.20	7.00	36.00	5.58	0.00	0.00	32.00	20.50	24.00
DBSTREAM@20	47.10	34.00	8.00	26.20	6.02	**30.50**	36.60	44.60	39.00	25.80
DenStream@20	**59.90**	**45.90**	15.10	23.56	8.75	11.40	37.60	57.80	58.30	**12.40**
RONDO@20	58.10	42.90	**21.10**	**44.70**	**4.03**	23.40	**38.70**	**65.40**	**63.50**	14.60
PAMAP2										
Time2State@20	41.10	23.70	1.10	**6.14**	**27.71**	50.10	40.30	34.90	32.60	6.20
DBSTREAM@20	55.00	52.10	**3.30**	4.71	38.57	59.60	52.90	49.50	47.00	19.70
DenStream@20	61.90	63.80	1.90	3.87	46.73	64.50	62.90	61.30	53.30	14.60
RONDO@20	**69.04**	**75.20**	1.17	3.44	59.57	**73.70**	**72.49**	**71.59**	**70.51**	**4.69**

Table 3: Cross-record transfer results on LARa and Drive&Act. RONDO-fit updates only MotifFormer on the target record.

Method	MoF	Recall	F1@50	Edit	SegRatio	Method	MoF	Recall	F1@50	Edit	SegRatio
LARa						**Drive&Act**					
Time2State	50.62	28.56	16.67	40.72	1.59	Time2State	44.80	17.53	4.99	14.07	14.37
E2USD	45.59	21.20	12.06	41.64	1.78	E2USD	43.75	14.94	6.00	13.34	14.68
RONDO	62.20	43.89	29.72	45.32	**1.24**	RONDO	54.27	38.11	8.97	**23.42**	**7.98**
RONDO-fit	**63.48**	**44.05**	**31.35**	**47.83**	1.25	RONDO-fit	**58.66**	**46.50**	**9.84**	18.94	9.28

temporal structure, with stronger segment-level quality and considerably less fragmentation than the streaming baselines.

This advantage becomes more apparent as additional observations accumulate: RONDO achieves stronger H-scores at later stages of the stream, suggesting that its unit and motif vocabularies can be progressively refined and expanded while retaining previously discovered structure. These results highlight the benefit of explicitly maintaining reusable temporal structure as the stream evolves, rather than treating continual discovery solely as online clustering.

Cross-record transfer of unit vocabularies. We further examine whether the discovered local units capture reusable temporal structure beyond the stream from which they are learned. We conduct this experiment on **LARa** and **Drive&Act**, both of which contain recordings from multiple participants and therefore provide a natural setting for evaluating transfer under changes in motif execution. Table 3 shows the cross-record transfer results. In the standard transfer setting, all methods learn from a source participant and directly apply the discovered structure to unseen target participants, a limited-data setting that tests how well their discovered patterns generalize across records. Under this setting, RONDO outperforms Time2State and E2USD on both datasets, suggesting that its discovered structure generalizes effectively across records. More importantly, our hierarchical formulation allows us to transfer only the learned unit vocabulary and discover new motifs from the target stream. This RONDO-fit variant further improves MoF and Recall on both datasets, indicating that local units remain reusable across records while higher-level motifs can be adapted to user-specific temporal compositions. This separation provides a path toward personalizing temporal structure discovery from limited target data without relearning the entire representation from scratch.

4.5 UNDERSTANDING RONDO

We further analyze RONDO from two perspectives: the contribution of its two main components and the effect of the underlying temporal representation.

Component ablation. We first evaluate the contribution of **UnitAlign** and **MotifFormer** by removing each component separately from the full approach. Removing UnitAlign tests the importance of adapting local representations to the recurring structure of the target stream, while removing MotifFormer eliminates explicit modeling of higher-level compositions. As shown in Table 4, the full model achieves the highest MoF across all four datasets, indicating that both components

Table 4: Ablation of RONDO. **w/o UA** removes UnitAlign while retaining MotifFormer, and **w/o MF** removes MotifFormer while retaining unit-level discovery.

Method	MoF	Recall	F1@50	Edit	SegRatio	Method	MoF	Recall	F1@50	Edit	SegRatio
OPPORTUNITY						**LARa**					
w/o MF	78.70	58.20	28.90	**49.48**	**1.13**	w/o MF	76.60	48.80	36.30	**47.66**	1.40
w/o UA	76.90	**65.40**	**35.00**	47.11	0.86	w/o UA	77.60	45.60	37.30	44.20	**1.26**
RONDO	**79.96**	59.19	30.59	48.06	1.23	RONDO	**78.44**	**53.01**	**38.50**	47.34	1.31
PAMAP2						**Drive&Act**					
w/o MF	80.30	92.60	11.12	12.94	17.99	w/o MF	55.90	35.99	7.66	13.12	11.69
w/o UA	82.43	**94.36**	22.31	**21.82**	**10.85**	w/o UA	66.28	55.84	**15.72**	**24.21**	**7.45**
RONDO	**84.12**	94.25	**22.91**	20.82	15.37	RONDO	**68.36**	**62.30**	14.29	22.22	8.04

Table 5: Representation sensitivity of RONDO across four datasets. Results compare TS2Vec, Time-MAE, and MantisV2 under the same discovery pipeline, showing that performance remains competitive across different self-supervised and pretrained temporal representations.

Method	MoF	Recall	F1@50	Edit	SegRatio	Method	MoF	Recall	F1@50	Edit	SegRatio
OPPORTUNITY						**LARa**					
TS2Vec	**79.96**	59.19	30.59	48.06	**1.23**	TS2Vec	78.44	53.01	38.50	47.34	**1.31**
Time-MAE	78.50	**65.30**	28.60	46.70	1.36	Time-MAE	**79.10**	**53.70**	**46.10**	57.30	2.14
MantisV2	72.30	61.00	**31.30**	**63.80**	2.40	MantisV2	74.60	51.50	39.90	**58.90**	2.23
PAMAP2						**Drive&Act**					
TS2Vec	**84.12**	94.25	**22.91**	**20.82**	**15.37**	TS2Vec	68.36	62.30	14.29	**22.22**	**8.04**
Time-MAE	83.50	95.30	16.10	16.30	17.40	Time-MAE	**75.10**	**68.10**	**15.40**	20.90	9.41
MantisV2	82.70	**97.30**	15.20	14.90	21.56	MantisV2	64.80	61.30	7.30	13.70	12.92

contribute to more consistent structure discovery. Removing MotifFormer degrades MoF across datasets and leads to substantial drops in segment-level performance on PAMAP2 and Drive&Act, highlighting the benefit of explicitly modeling compositions of reusable units. Removing UnitAlign particularly reduces discovery quality on LARa and Drive&Act, while the less uniform effects on F1@50 and Edit suggest that representation adaptation primarily improves recurring-structure recovery rather than boundary estimation.

Representation backbone. We next examine the sensitivity of RONDO to the choice of temporal representation encoder. We test the encoder within UnitAlign to TS2Vec (Yue et al., 2022), Time-MAE (Cheng et al., 2026), and MantisV2 Feofanov et al. (2026) while keeping the remaining discovery pipeline unchanged. As shown in Table 5, performance remains generally stable across representation choices, although the strongest encoder varies by dataset. TS2Vec achieves the highest MoF on OPPORTUNITY and PAMAP2, while Time-MAE performs best on Drive&Act and LARa and improves several segment-level metrics. MantisV2 is competitive on some segmentation metrics but shows less consistent discovery performance overall. These results suggest that RONDO is not tied to a particular representation model, while the quality of the underlying representation still influences the discovered temporal structure.

5 CONCLUSION

We introduced RONDO, an unsupervised approach for discovering multiscale recurring structure in continuous temporal streams. By explicitly modeling reusable local units and their higher-level compositions, RONDO captures recurring structure beyond independent states or subsequences and continually updates its vocabularies as the stream evolves. Experiments across diverse datasets demonstrate improved discovery and temporal segmentation over existing unsupervised discovery methods. These results highlight the benefits of modeling temporal recurrence as compositional and evolving structure, and motivate extending it to richer temporal hierarchies in future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with dataset preprocessing and reformatting, and with implementing and debugging code for baseline experiments. This included adapting released baseline implementations to our data format, writing evaluation and scoring scripts, and developing scripts for running and aggregating experiments. We also used generative AI tools to improve the clarity and readability of portions of the manuscript.

We did not use generative AI tools to design the proposed method, formulate research hypotheses, design the experimental methodology, interpret experimental results, or generate research ideas.

We reviewed and verified all AI-assisted work before inclusion in the paper. In particular, reported results were cross-checked against saved configurations and per-recording outputs, baseline settings were checked against the corresponding released implementations, and modifications to released baseline implementations are documented in the appendix. All AI-assisted text was reviewed and edited by the authors.

ETHICS STATEMENT

This work uses publicly available datasets and does not involve new data collection or human-subject studies. We follow the usage terms of the respective datasets and use the data solely for research purposes. Our method performs unsupervised temporal structure discovery and does not infer sensitive personal attributes. We are not aware of additional ethical concerns specific to this work beyond those associated with the original datasets and their collection protocols.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, preprocessing pipeline, evaluation protocol, and experimental settings in the main paper and appendix. The appendix includes dataset-specific preprocessing and windowing settings, method hyperparameters, metric definitions, implementation details, and all modifications made to released baseline implementations. We also report results across multiple random seeds where applicable. Code and scripts for reproducing the experiments will be released following the review process.

REFERENCES

Charu C Aggarwal, S Yu Philip, Jiawei Han, and Jianyong Wang. A framework for clustering evolving data streams. In *Proceedings 2003 VLDB conference*, pp. 81–92. Elsevier, 2003.

Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):1–33, 2014.

Feng Cao, Martin Estert, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM international conference on data mining*, pp. 328–339. SIAM, 2006.

Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34(15):2033–2042, 2013.

Jiuxu Chen, Nupur Thakur, Sachin Chhabra, and Baoxin Li. Unsupervised action anticipation through action cluster prediction. *IEEE Open Journal of Signal Processing*, 6:641–650, 2025.

Mingyue Cheng, Xiaoyu Tao, Zhiding Liu, Qi Liu, Hao Zhang, Rujiao Zhang, and Enhong Chen. Timemae: Self-supervised representations of time series with decoupled masked autoencoders. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pp. 498–508, 2026.

Bill Chiu, Eamonn Keogh, and Stefano Lonardi. Probabilistic discovery of time series motifs. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 493–498, 2003.

Hao Dai and Jagmohan Chauhan. Continual generalized category discovery: Learning and forgetting from a bayesian perspective. *arXiv preprint arXiv:2507.17382*, 2025.

Arik Ermshaus, Patrick Schäfer, and Ulf Leser. Clap–state detection from time series. *arXiv preprint arXiv:2504.01783*, 2025.

Vasilii Feofanov, Songkang Wen, Jianfeng Zhang, Lujia Pan, and Ievgen Redko. Mantisv2: Closing the zero-shot gap in time series classification with synthetic data and test-time strategies. *arXiv preprint arXiv:2602.17868*, 2026.

Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. How shall we evaluate egocentric action recognition? In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 2373–2382. IEEE, 2017.

Dayoung Gong, Joonseok Lee, Deunsol Jung, Suha Kwak, and Minsu Cho. Activity grammars for temporal action segmentation. *Advances in Neural Information Processing Systems*, 36:75413–75433, 2023.

Michael Hahsler and Matthew Bolaños. Clustering data streams based on shared density between micro-clusters. *IEEE transactions on knowledge and data engineering*, 28(6):1449–1461, 2016.

David Hallac, Sagar Vare, Stephen Boyd, and Jure Leskovec. Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 215–223, 2017.

Shruthi K Hiremath, Yasutaka Nishimura, Sonia Chernova, and Thomas Plötz. Bootstrapping human activity recognition systems for smart homes from scratch. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 6(3):1–27, 2022.

Yiding Jiang, Evan Liu, Benjamin Eysenbach, J Zico Kolter, and Chelsea Finn. Learning options via compression. *Advances in Neural Information Processing Systems*, 35:21184–21199, 2022.

Kouki Kawabata, Yasuko Matsubara, and Yasushi Sakurai. Streamscope: Automatic pattern discovery over data streams. In *Proceedings of the First International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*, pp. 1–8, 2018.

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pp. 3418–3428. PMLR, 2019.

Mahesh Venkata Krishna, Paul Bodesheim, Marco Körner, and Joachim Denzler. Temporal video segmentation by event detection: A novelty detection approach. *Pattern recognition and image analysis*, 24(2):243–255, 2014.

Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 780–787, 2014.

Sateesh Kumar, Sanjay Haresh, Awais Ahmed, Andrey Konin, M Zeeshan Zia, and Quoc-Huy Tran. Unsupervised action segmentation by joint representation learning and online clustering. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20142–20153. IEEE, 2022.

Zhichen Lai, Huan Li, Dalin Zhang, Yan Zhao, Weizhu Qian, and Christian S Jensen. E2usd: Efficient-yet-effective unsupervised state detection for multivariate time series. In *Proceedings of the ACM Web Conference 2024*, pp. 3010–3021, 2024.

Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *European conference on computer vision*, pp. 36–52. Springer, 2016.

Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1003–1012. IEEE, 2017.

Yuan Li, Jessica Lin, and Tim Oates. Visualizing variable-length time series motifs. In *Proceedings of the 2012 SIAM international conference on data mining*, pp. 895–906. SIAM, 2012.

Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn Keogh. Matrix profile x: Valmod-scalable discovery of variable-length motifs in data series. In *Proceedings of the 2018 international conference on management of data*, pp. 1053–1066, 2018.

Jiexi Liu and Songcan Chen. Timesurl: Self-supervised contrastive learning for universal time series representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 13918–13926, 2024.

Shijie Ma, Fei Zhu, Zhun Zhong, Wenzhuo Liu, Xu-Yao Zhang, and Cheng-Lin Liu. Happy: A debiased learning framework for continual generalized category discovery. *Advances in Neural Information Processing Systems*, 37:50850–50875, 2024.

Amadeus Maes, Mauricio Barahona, and Claudia Clopath. Learning compositional sequences with multiple time scales through a hierarchical network of spiking neurons. *PLoS computational biology*, 17(3):e1008866, 2021.

Manuel Martin, Alina Roitberg, Monica Haurilet, Matthias Horne, Simon Reiß, Michael Voit, and Rainer Stiefelhagen. Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles. In *2019 IEEE/CVF international conference on computer vision (ICCV)*, pp. 2801–2810. IEEE, 2019.

Yasuko Matsubara, Yasushi Sakurai, and Christos Faloutsos. Autoplait: Automatic mining of co-evolving time sequences. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 193–204, 2014.

David Minnen, Thad Starner, Irfan Essa, and Charles Isbell. Discovering characteristic actions from on-body sensor data. In *2006 10th IEEE international symposium on wearable computers*, pp. 11–18. IEEE, 2006.

David Minnen, Charles L Isbell, Irfan Essa, and Thad Starner. Discovering multivariate motifs using subsequence density estimation and greedy mixture learning. In *Proceedings of the national conference on artificial intelligence*, volume 22, pp. 615. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

Abdullah Mueen and Eamonn Keogh. Online discovery and maintenance of time series motifs. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1089–1098, 2010.

Abdullah Mueen, Eamonn Keogh, Qiang Zhu, Sydney Cash, and Brandon Westover. Exact discovery of time series motifs. In *Proceedings of the 2009 SIAM international conference on data mining*, pp. 473–484. SIAM, 2009.

Scott Niekum, Sarah Osentoski, George Konidaris, and Andrew G Barto. Learning and generalization of complex tasks from unstructured demonstrations. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5239–5246. IEEE, 2012.

Friedrich Niemann, Christopher Reining, Fernando Moya Rueda, Nilah Ravi Nair, Janine Anika Steffens, Gernot A Fink, and Michael Ten Hompel. Lara: Creating a dataset for human activity recognition in logistics using semantic attributes. *Sensors*, 20(15):4083, 2020.

Keon-Hee Park, Hakyung Lee, Kyungwoo Song, and Gyeong-Moon Park. Online continuous generalized category discovery. In *European Conference on Computer Vision*, pp. 53–69. Springer, 2024.

Jizong Peng, Marco Pedersoli, and Christian Desrosiers. Mutual information deep regularization for semi-supervised segmentation. In *Medical imaging with deep learning*, pp. 601–613. PMLR, 2020.

Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*, pp. 108–109. IEEE, 2012.

Fernando Moya Rueda and Gernot A Fink. From human pose to on-body devices for human-activity recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 10066–10073. IEEE, 2021.

Marta Sanzari, Valsamis Ntouskos, and Fiora Pirri. Discovery and recognition of motion primitives in human activities. *PloS one*, 14(4):e0214499, 2019.

Federico Spurio, Emad Bahrami, Gianpiero Francesca, and Juergen Gall. Hierarchical vector quantization for unsupervised action segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 6996–7005, 2025.

Federico Spurio, Emad Bahrami, Olga Zatsarynna, Yazan Abu Farha, Gianpiero Francesca, and Juergen Gall. Looking into the unknown: Exploring action discovery for segmentation of known and unknown actions. *Computer Vision and Image Understanding*, pp. 104849, 2026.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. *arXiv preprint arXiv:2201.02609*, 2022.

Chengyu Wang, Kui Wu, Tongqing Zhou, and Zhiping Cai. Time2state: An unsupervised framework for inferring the latent states in time series data. *Proceedings of the ACM on Management of Data*, 1(1):1–18, 2023.

Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16544–16554. IEEE, 2023.

Shuchen Wu, Noémi Élteto, Ishita Dasgupta, and Eric Schulz. Learning structure from the ground up—hierarchical representation learning by chunking. *Advances in Neural Information Processing Systems*, 35:36706–36721, 2022.

Yanan Wu, Zhixiang Chi, Yang Wang, and Songhe Feng. Metagcd: Learning to continually learn in generalized category discovery. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1655–1665. IEEE, 2023.

Ming Xu and Stephen Gould. Temporally consistent unbalanced optimal transport for unsupervised action segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14618–14627. IEEE, 2024.

Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pp. 1317–1322. Ieee, 2016.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 8980–8987, 2022.

Mi Zhang, Wenyao Xu, Alexander A Sawchuk, and Majid Sarrafzadeh. Sparse representation for motion primitive-based human activity modeling and recognition using wearable sensors. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pp. 1807–1810. IEEE, 2012.

Feng Zhou, Fernando De la Torre, and Jessica K Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):582–596, 2012.

APPENDIX

A QUALITATIVE ANALYSIS

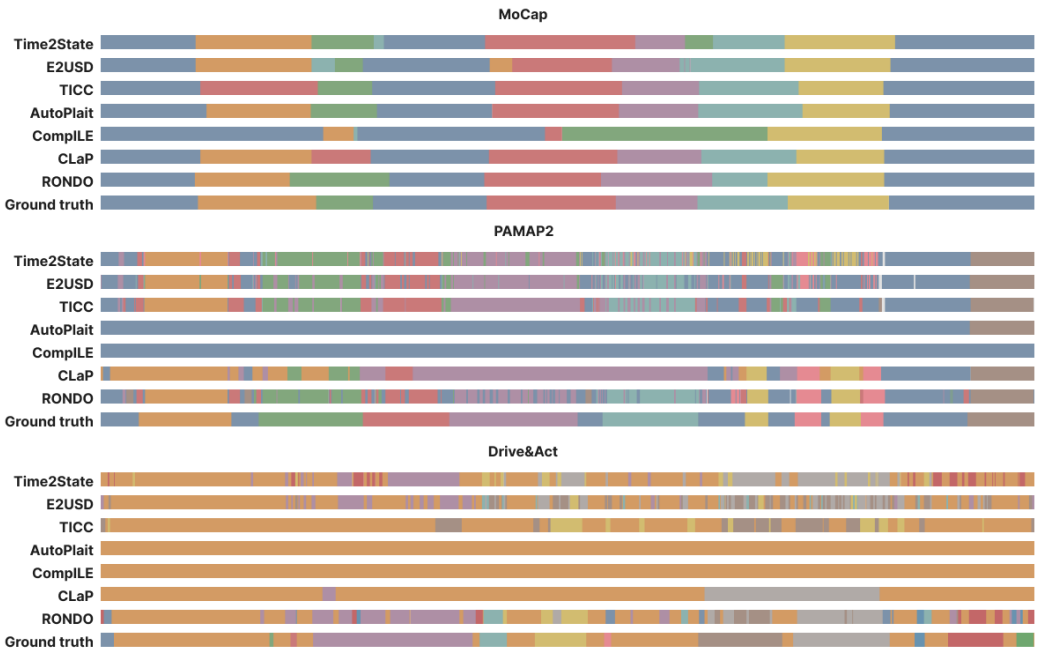


Figure 3: Segmentation of one recording from three representative datasets, against the ground truth (bottom). Predicted states are coloured by the class they map to; a single-colour row means every predicted state mapped to the same class.

Beyond the quantitative metrics in the main text, we randomly sample recordings for qualitative inspection. Figure 3 shows that baseline discovery often struggles to balance local variation and long-range consistency: overly sensitive state changes lead to fragmented segments, whereas aggressive grouping merges distinct activities or collapses large portions of a sequence into a few states. In contrast, RONDO generally preserves clearer temporal structure while still capturing short activity transitions, although some long activities remain split into multiple recurring units.

We further inspect the relationship between the two hierarchical levels through their activity purity. Across most recordings, discovered units are temporally persistent, with around 75% to over 90% of adjacent windows retaining the same unit assignment. The unit vocabulary is also repeatedly reused within each recording, and in several datasets individual units already show relatively strong alignment with activity classes. This suggests that recurring local structure alone can often provide meaningful discrimination, particularly when activities exhibit distinctive and stable local dynamics.

The motif layer becomes more useful when this local alignment is weaker. On datasets where units are shared across several activities, motif-level grouping can improve activity consistency by roughly a few to more than ten percentage points, indicating that longer temporal context helps disambiguate locally similar patterns. In contrast, when unit-level purity is already high, motif composition provides only marginal gains, especially for short actions that are shorter than the modeled motif scale. These observations suggest that the hierarchy is not simply adding another clustering layer: units capture reusable local primitives, while motifs provide contextual disambiguation mainly when local patterns alone are insufficient.

B Evaluation Metrics

For completeness, we provide the definitions of all evaluation metrics used in the paper. These metrics evaluate discovery quality, temporal segmentation, and continual adaptation from complementary perspectives.

B.1 Temporal Alignment and Label Mapping

All predictions and annotations are projected onto a common temporal axis consisting of consecutive intervals. Let y_t denote the ground-truth class label and $\hat{z}_t$ the discovered cluster at interval t. Ground-truth labels are used only for evaluation and are not involved in training, clustering, or online updating. Because different execution-specific compositions may correspond to the same semantic class, we use a many-to-one mapping from discovered clusters to ground-truth class labels, as illustrated in Fig. 4. For each discovered cluster c, we assign the label with the largest duration-weighted overlap:

$$\pi(c) = \arg\max_{y} \sum_{t} \Delta_t \mathbb{I}[\hat{z}_t = c \wedge y_t = y], \tag{7}$$

where Δ_t is the duration of interval t. The mapped prediction is $\tilde{y}_t = \pi(\hat{z}_t)$. The mapping is estimated separately for each recording on the evaluation set and fixed across all continual stages. This evaluation-only many-to-one protocol follows prior over-clustering and action-cluster evaluation settings (Peng et al., 2020; Chen et al., 2025).

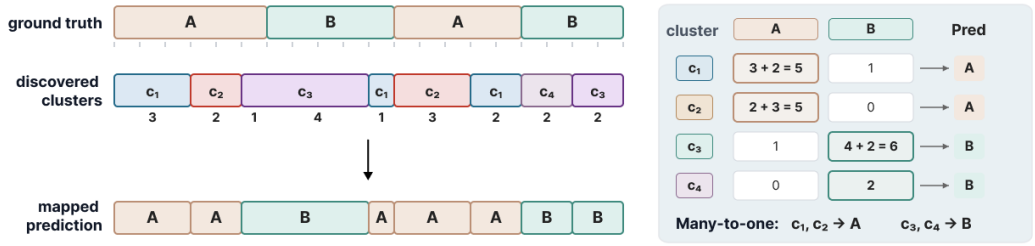


Figure 4: Illustration of the evaluation-only many-to-one mapping. Discovered clusters are matched to ground-truth labels by their duration-weighted overlap on a common temporal axis, allowing multiple clusters to map to the same semantic label.

B.2 Class and Segment Metrics

We report duration-weighted mean over frame (MoF), discovery recall, Edit score, and segmental F1. MoF is computed as

$$\text{MoF} = \frac{\sum_{t \in \mathcal{T}*\text{eval}} \Delta_t, \mathbb{I}[\tilde{y} * t = y_t]}{\sum *t \in \mathcal{T} * \text{eval}\Delta_t}. \tag{8}$$

Discovery recall measures the proportion of ground-truth class instances that are successfully discovered:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{9}$$

For the Edit score, consecutive intervals with the same mapped label are first merged into segments. Let S and $\hat{S}$ be the ground-truth and predicted segment-label sequences. We compute

$$\text{Edit} = 100 \left(1 - \frac{d_{\text{Lev}}(\hat{S}, S)}{\max(|\hat{S}|, |S|)} \right), \tag{10}$$

where d_{Lev} is the Levenshtein distance (Lea et al., 2016).

For segmental F1, a predicted segment p matches a ground-truth segment g when they have the same mapped label and

$$\text{IoU}(p, g) = \frac{|p \cap g|}{|p \cup g|} \geq \tau. \tag{11}$$

We report

$$\text{F1@}\tau = \frac{2P_\tau R_\tau}{P_\tau + R_\tau}, \qquad \tau \in 0.10, 0.25, 0.50, \tag{12}$$

following standard temporal action segmentation practice (Lea et al., 2017). Segment matching is one-to-one, so duplicate predictions do not receive repeated credit.

B.3 Continual Discovery Metrics

We evaluate the model after each update stage using performance on previously known classes, newly discovered classes, and all classes. Let $A^{(s)}_{\text{old}}$ and $A^{(s)}_{\text{new}}$ denote the corresponding class-level performance at stage s.

The H-score measures the balance between retaining old classes and discovering new ones:

$$H^{(s)} = \frac{2A^{(s)}_{\text{old}} A^{(s)}_{\text{new}}}{A^{(s)}_{\text{old}} + A^{(s)}_{\text{new}}}. \tag{13}$$

Forgetting is measured as the degradation of old-class performance relative to the initial model:

$$M_f^{(s)} = A^{(0)}_{\text{old}} - A^{(s)}_{\text{old}}. \tag{14}$$

We report the final forgetting across stages. Lower M_f indicates better retention of previously learned classes (Vaze et al., 2022; Wen et al., 2023; Wu et al., 2023; Park et al., 2024).

C Datasets and Preprocessing Details

C.1 Overview

We evaluate RONDO on six public datasets spanning motion capture, wearable inertial sensing, multimodal body sensing, and video-derived human pose. Importantly, these datasets also cover substantially different behavioral domains, including controlled human motion (MoCap), everyday physical and daily-living activities (PAMAP2 and ActRecTut), in-vehicle activities (Drive&Act), sensor-rich daily routines (OPPORTUNITY), and industrial workplace activities (LARa). Together, they provide substantial variation in sensing modality, sampling rate, dimensionality, sequence length, participant population, and temporal organization, allowing us to evaluate whether recurring temporal structure can be discovered consistently across heterogeneous domains rather than within a single class setting.

Each recording is treated as an independent continuous temporal stream, and discovery is performed without using class annotations or the number of ground-truth classes during training. Ground-truth labels are used only for evaluation after discovery. Unless otherwise stated, segmentation metrics are computed independently within each recording and then aggregated across recordings, avoiding artificial transitions across recording boundaries. As summarized in Table 6, the benchmark contains 267 recordings and approximately 7.45 million frames in total.

Table 6: Summary of the six evaluation datasets. *Rec.* denotes the number of independent recordings over which temporal segmentation is evaluated. *Raw ch.* denotes the dimensionality of the original input signals before method-specific preprocessing; the actual input dimensionality used by each method is reported in Table 7.

Dataset	Source	Subj.	Rec.	Frames	Rate (Hz)	Raw ch.	Classes
MoCap	CMU MoCap, subject 86	1	9	65,611	—	4	9
PAMAP2	PAMAP2 Protocol	8	8	2,864,056	100	40	13
ActRecTut	ActRecTut	2	4	200,693	42	24	17
Drive&Act	Drive&Act, OpenPose 3D	15	29	1,222,442	30	104	13
OPPORTUNITY	OPPORTUNITY UCI	4	24	869,387	30.3	242	5
LARa	LARa v3, MbientLab IMU	8	193	2,224,452	100	30	8
Total			**267**	**7,446,641**			

C.2 Preprocessing and input dimensionality

We convert all datasets into a common representation consisting of continuous multivariate recordings, annotation track, and recording boundaries. This provides every method with the same temporal organization and evaluation interface despite differences in the original data formats. All analysis windows are restricted to individual recordings. This prevents artificial transitions from being introduced by concatenating temporally unrelated sequences. More important, ground-truth annotations are retained strictly as evaluation metadata. All discovery methods operate without access to class labels, class identities, or the number of ground-truth classes; annotations are exposed only to the evaluation procedure after discovery.

The datasets vary substantially in input dimensionality, ranging from 4 to 242 raw channels. Because several classical discovery methods become difficult to apply directly in high-dimensional spaces, we use dimensionality reduction where necessary while preserving each method's standard representation pipeline. Time2State and E2USD include learned representation encoders and therefore receive the native input channels whenever computationally tractable. For TICC, AutoPlait, and CLaP, higher-dimensional datasets are projected with PCA using a transformation fitted once per dataset and shared across all recordings.

We additionally verify that these dimensionality choices do not drive the reported differences. For Time2State and E2USD, we repeat the experiments using the corresponding PCA-reduced inputs and obtain results comparable to those with native-dimensional inputs; we therefore report the native-input results. For methods using PCA, we evaluate multiple projection dimensionalities and verify that increasing the retained dimensionality beyond the selected value does not yield meaningful improvements. Thus, the selected projections retain sufficient information for each method rather than imposing an artificially low-dimensional bottleneck.

OPPORTUNITY contains 242 input channels and is high-dimensional even for the encoder-based methods; we therefore use the same 54-dimensional PCA projection for all methods, corresponding to the minimum number of components retaining 90% of the variance. MoCap is already represented by four channels and requires no reduction. For PAMAP2, we follow the commonly used subset of nine $\pm 16, g$ accelerometer channels from the hand, chest, and ankle sensors. Table 7 summarizes the resulting preprocessing settings.

D Implementation Details

We provide additional implementation details for the RONDO pipeline used throughout the paper. Unless otherwise noted, all hyperparameters are fixed across datasets. Class annotations are never used to optimize any component of the model and are introduced only after motif discovery for post-hoc semantic mapping and evaluation.

Table 7: Preprocessing settings for each dataset. *Raw* is the original channel count; the next three columns give the input channels each method receives (native, the published channel selection, or the number of PCA components). The window/step applies to the sliding-window methods.

Dataset	Raw	Time2State, E2USD	TICC, AutoPlait	CLaP, CompILE	Window / step
MoCap	4	4 (native)	4 (native)	4 (native)	256 / 50
PAMAP2	40	9 (selection)	9 (selection)	9 (selection)	200 / 100
ActRecTut	24	24 (native)	10 (PCA)	8 (PCA)	90 / 30
Drive&Act	104	104 (native)	16 (PCA)	16 (PCA)	90 / 30
OPPORTUNITY	242	54 (PCA)	54 (PCA)	54 (PCA)	30 / 15
LARa	30	30 (native)	16 (PCA)	16 (PCA)	200 / 50

D.1 Temporal Representation Encoders

We used three temporal representation encoders in evaluation: TS2Vec, Time-MAE, and MantisV2. Each encoder converts a sliding window into a fixed-dimensional representation that is subsequently processed by UnitAlign. The encoder parameters are not updated during subsequent static discovery or continual vocabulary adaptation.Unless stated otherwise, we follow their respective official implementations and default architectures.

TS2Vec. We train a separate TS2Vec model from scratch for each recording. We set the output representation dimension to 128, use a batch size of 64, and train for 100 optimization iterations. The resulting 128-dimensional embedding is used as the input to UnitAlign.

Time-MAE. We train a separate Time-MAE model from scratch for each recording. Each input channel is standardized using statistics computed over all windows and time steps within that recording. We use a patch size of 2 samples, a four-layer Transformer encoder, a two-layer regressor, dropout 0.1, and batch size 64. Models are trained for 10 epochs. At inference, the encoder tokens are mean-pooled to obtain one 64-dimensional representation per window.

MantisV2. We use the publicly released pretrained MantisV2 checkpoint without target-domain fine-tuning or external input normalization. Each window is linearly resized to length 512 before encoding. For feature extraction, we use the third Transformer layer (`return_transf_layer=2`) with the combined CLS and mean-token representation, yielding a 512-dimensional embedding for each univariate input. For multivariate data, channels are encoded independently and their embeddings are averaged to obtain one 512-dimensional representation per window.

Although the encoders produce representations of different dimensionalities, UnitAlign maps all encoder outputs into the same latent space before subsequent discovery. Therefore, the downstream model capacity is held fixed across encoder variants.

D.2 UnitAlign Implementation

UnitAlign adapts the representations produced by the temporal encoder to the recurring local structure of the target stream. Let $y \in \mathbb{R}^D$ denote an encoder representation. We use a residual bottleneck adaptor

$$h = \mathrm{LayerNorm}\left(y + \alpha W_2\left(\mathrm{Dropout}\left(\mathrm{GELU}(W_1 y)\right)\right)\right), \tag{15}$$

followed by

$$z = \mathrm{normalize}(W_h h). \tag{16}$$

W_1 projects the encoder representation into a bottleneck and W_2 maps it back to the encoder dimension. We initialize W_2 to zero so that the residual adaptor starts close to an identity transformation. A final projection W_h maps the adapted features to a shared ℓ_2-normalized embedding space.

Geometry-preserving initialization. Before introducing discrete unit assignments, we first train the adaptor to preserve the geometry of the encoder representation while avoiding representation collapse. The objective is

$$\mathcal{L}_{\mathrm{init}} = \lambda_{\mathrm{geom}}\,\mathcal{L}_{\mathrm{geom}} + \lambda_{\mathrm{VIC}}\,\mathcal{L}_{\mathrm{VIC}}, \tag{17}$$

where $\mathcal{L}_{\text{geom}}$ preserves the pairwise geometry of the original representation space:

$$\mathcal{L}_{\text{geom}} = \frac{1}{B^2} \|S(Y) - S(Z)\|_F^2 , \tag{18}$$

with $S(\cdot)$ denoting the pairwise cosine-similarity matrix within a minibatch. $\mathcal{L}_{\text{VIC}}$ combines a variance hinge that discourages dimension-wise collapse with an off-diagonal covariance penalty that reduces feature redundancy, both applied to the pre-projection adapted features.

Iterative unit refinement. After initialization, UnitAlign alternates between unsupervised clustering and representation refinement. At iteration k, DeepDPM is fitted to the current adapted representations z_i to obtain pseudo unit assignments $\hat{u}_i$. The adaptor is then refined using

$$\mathcal{L}_{\text{refine}} = \lambda_{\text{sup}}\mathcal{L}_{\text{SupCon}} + \lambda_{\text{geom}}\mathcal{L}_{\text{geom}} + \lambda_{\text{VIC}}\,\mathcal{L}_{\text{VIC}}. \tag{19}$$

The supervised contrastive term treats windows assigned to the same discovered component as positives, using only DeepDPM assignments as pseudo-labels and no semantic annotations. Samples identified as noise by the clustering model are excluded. The variance and covariance regularizers are applied to the pre-projection adapted features, while the geometry term continues to preserve the structure of the original encoder space.

Clustering and adaptor refinement are alternated until the discovered unit assignments stabilize, as measured by the ARI between consecutive iterations, or until the maximum number of refinement rounds is reached. After refinement, DeepDPM is fitted once more to the final adapted representations to obtain the unit vocabulary. Each retained component defines a local unit, represented by the mean adapted representation of its members. The resulting unit assignments are temporally smoothed before each window is replaced by its corresponding unit centroid, forming the centroid timeline used for motif discovery.

Temporal smoothing. The unit assignments produced by DeepDPM are further refined with Viterbi decoding to discourage implausibly frequent transitions between neighboring windows. The emission cost is computed in the DeepDPM latent space from the distance between each window representation and the corresponding unit center, while a transition penalty discourages changes in unit identity across adjacent windows. The transition scale is determined from the typical embedding change between consecutive windows. The resulting smoothed unit sequence is used to construct the centroid timeline for subsequent motif discovery.

D.3 MotifFormer Implementation

MotifFormer operates on the centroid timeline produced by UnitAlign and learns representations of recurring compositions of local units across multiple temporal scales. For a subsequence, each unit centroid is first projected into the Transformer feature space. We prepend a learnable `[CLS]` token and add both positional and subsequence-length embeddings. The resulting sequence is processed by a Transformer encoder, and the output of the `[CLS]` token is used as the descriptor of the complete subsequence.

Multi-view motif representation learning. To encourage descriptors to capture the underlying recurring composition rather than individual units, we construct two corrupted views of each subsequence by masking and dropping unit tokens. MotifFormer is trained with the objective

$$\mathcal{L}_{\text{motif}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{view}}\mathcal{L}_{\text{view}} + \lambda_{\text{cross}}\mathcal{L}_{\text{cross}} + \lambda_{\text{pseudo}}\mathcal{L}_{\text{pseudo}} + \lambda_{\text{var}}\mathcal{L}_{\text{var}}. \tag{20}$$

$\mathcal{L}_{\text{rec}}$ reconstructs the original unit-centroid representations at corrupted positions from the contextualized Transformer features. $\mathcal{L}_{\text{view}}$ is a contrastive objective that aligns the `[CLS]` descriptors of two corrupted views of the same subsequence. $\mathcal{L}_{\text{cross}}$ further aligns subsequences that begin at the same temporal position but span different lengths, encouraging MotifFormer to capture structure that is consistent across temporal scales.

To introduce coarse structural supervision without semantic labels, $\mathcal{L}_{\text{pseudo}}$ predicts periodically refreshed clusters of subsequence descriptors. These pseudo-labels are obtained entirely from the current representation space; only high-confidence assignments are retained for this term. Finally, $\mathcal{L}_{\text{var}}$ prevents collapse by encouraging sufficient feature variation across a minibatch.

Soft coverage across temporal scales. Motif discovery proceeds from longer to shorter subsequence lengths. After motifs at a given length have been identified, their accepted intervals are recorded on the timeline and used to reweight candidates at shorter scales. For a candidate subsequence q, we compute its coverage ratio

$$\rho(q) = \frac{|\mathcal{I}(q) \cap \mathcal{C}|}{|\mathcal{I}(q)|}, \tag{21}$$

where $\mathcal{I}(q)$ denotes the temporal interval spanned by q and $\mathcal{C}$ is the union of intervals already claimed by longer motifs. Its sampling weight is then

$$w(q) = 1 - \rho(q). \tag{22}$$

Thus, a completely uncovered candidate retains full weight, whereas candidates that largely overlap previously discovered structure contribute less to clustering without being removed entirely.

Motif matching and multi-length fusion. At inference time, the smoothed unit sequence is converted into the same centroid timeline used during motif discovery. MotifFormer then encodes all candidate subsequences over the supported temporal lengths. For each candidate, we evaluate its compatibility with every motif discovered at the corresponding length.

A candidate is accepted only when it satisfies both a distance and a posterior criterion in the DeepDPM latent space. Specifically, its distance to the motif center must fall within the motif-specific acceptance radius, and its posterior probability under the corresponding mixture component must exceed the stored confidence threshold. Both statistics are estimated from the motif members observed during discovery.

For an accepted match between candidate q and motif m, we define a normalized distance

$$r(q, m) = \frac{d(q, \mu_m)}{R_m}, \tag{23}$$

where μ_m and R_m denote the latent center and acceptance radius of motif m, respectively. Smaller values therefore indicate stronger matches. Within each subsequence length, if multiple candidates associate the same window with the same motif, we retain only the match with the smallest normalized distance.

Predictions from different temporal lengths are then fused at the motif level. Let $\mathcal{L}_{t,m}$ denote the set of lengths that support motif m at window t, and let $r_{t,m}^{(L)}$ be the best normalized distance contributed by length L. We compute

$$s_{t,m} = \left(\frac{1}{|\mathcal{L}_{t,m}|} \sum_{L \in \mathcal{L}_{t,m}} \exp\left[-\min\left(r_{t,m}^{(L)}, \tau_r \right) \right] \right) \cdot b(|\mathcal{L}_{t,m}|)\,, \tag{24}$$

where τ_r limits the influence of weak matches and $b(\cdot)$ provides a bounded bonus when the same motif is supported by multiple temporal lengths. The motif with the highest fused score is selected for each window. Windows that are not accepted by any motif at any temporal length are assigned to OTHER.

D.4 CONTINUAL UPDATE IMPLEMENTATION

During continual discovery, each incoming window is first encoded using the existing temporal encoder and UnitAlign adaptor. At the unit level, the incoming window is matched against the existing unit vocabulary using the same distance-radius and posterior-confidence criteria described above. If the window is accepted by an existing unit, its representation is used to incrementally update the corresponding unit prototype and distribution statistics.

The updated unit timeline is then used to construct subsequences ending around the newly arrived window. These subsequences are encoded by MotifFormer and matched against the existing motif vocabulary using the motif-specific distance and posterior criteria. Accepted subsequences update the corresponding motif prototype and its matching statistics.

Data that cannot be confidently explained by the current vocabulary are not immediately assigned to a new entry. Instead, rejected windows at the unit level and rejected subsequences at the motif level are placed into separate novelty buffers. This separation allows the model to accumulate evidence for persistent new structure before modifying either vocabulary.

Micro-cluster evolution. Each novelty buffer is organized as a set of online micro-clusters. A rejected point is either absorbed by a nearby micro-cluster or used to initialize a new one. Each micro-cluster maintains a prototype and an effective support that is updated as new points arrive and decays over time, allowing transient or isolated novelty to disappear without entering the vocabulary.

A micro-cluster becomes eligible for promotion only after accumulating sufficient support and exhibiting persistent temporal recurrence. For unit novelty, candidate points are further consolidated in the existing DeepDPM latent space so that promotion is based on coherent structure rather than individual rejected windows. We additionally require the candidate to span a sufficient temporal extent and to recur across multiple temporal regions, preventing a short local burst from being interpreted as a new unit.

Once a candidate micro-cluster satisfies the promotion criteria, we compare its prototype with the existing vocabulary. If it remains compatible with a nearby existing entry, the accumulated data are merged into that entry and its prototype and distribution statistics are refined. Otherwise, the candidate is promoted to a new vocabulary entry and appended to the unit vocabulary.

The motif-level novelty buffers follow the same principle independently at each temporal length. Rejected subsequences accumulate into length-specific micro-clusters; persistent clusters are compared with existing motifs at the same scale and are either merged with a compatible motif or promoted as a new motif. Consequently, both vocabularies can be continuously *refined* when new points represent variations of known structure and *expanded* when persistent previously unseen units or motifs emerge.

D.5 Implementation Hyperparameters

We summarize the key hyperparameters used in our experiments in Table 8.

E Baseline Implementation Details

E.1 Overview

We compare against six representative methods for unsupervised temporal segmentation and state discovery. All baselines are run using the authors' released implementations, preserving their original learning and inference procedures. We modify only the data-loading and input/output interfaces required to apply each method to our datasets.

All methods are converted to a common output format consisting of one predicted state label per timestamp at the original timeline resolution. Metrics are then computed once using the same evaluation code for every method. Metrics are defined in Appendix B. Each metric is computed separately for every recording and then averaged across recordings, except SegRatio, which is computed from the total numbers of predicted and ground-truth segments over the dataset. We report results on the intersection of recordings for which all methods produced valid outputs.

E.2 Experimental Adaptation and Compute

For AutoPlait, TICC, Time2State, E2USD, DenStream, and DBStream, we use the authors' released implementations without modifying the underlying algorithms, using the hyperparameters specified in Appendix E.3. DenStream and DBStream are used only in the continual-discovery experiments and follow their original online update procedures. For scale-sensitive methods: **TICC**, **CLaP**, and **CompILE**, we standardize each input channel to zero mean and unit variance within each recording. This avoids differences in channel scale disproportionately affecting TICC's covariance-based state modeling (Hallac et al., 2017), CLaP's classification of temporal subsequences (Ermshaus et al., 2025), and, in our continuous-input adaptation of CompILE, the reconstruction objective used to

Table 8: Key hyperparameters used in RONDO. Unless otherwise stated, all settings are fixed across datasets.

Component	Setting	Value
UnitAlign	bottleneck / output dimension	256 / 128
	initialization epochs	20
	$\lambda_{\text{geom}}, \lambda_{\text{VIC}}$	0.2 / 0.05
	refinement epochs / max rounds	3 / 20
	$\lambda_{\text{sup}}, \lambda_{\text{var}}, \lambda_{\text{cov}}, \lambda_{\text{geom}}$	1.0 / 1.0 / 0.04 / 0.25
	SupCon temperature	0.1
	convergence	ARI ≥ 0.995, patience 2
	Viterbi transition scale	2.0
DeepDPM	latent dimension	32
	AE training epochs	10
	DP concentration	1.0
	unit / motif component cap	30 / 36
	minimum component weight	0.01
MotifFormer	lengths	2–10
	Transformer	1 layer, $d = 128$, 4 heads
	training epochs	20
	mask / token-drop probability	0.15 / 0.10
	$\lambda_{\text{rec}}, \lambda_{\text{view}}, \lambda_{\text{cross}}, \lambda_{\text{pseudo}}, \lambda_{\text{var}}$	0.1 / 1.0 / 0.5 / 0.5 / 0.01
	contrastive temperature	0.1
Motif discovery	coverage declaration	strongest 10%
	distance / posterior threshold	P95 / P5
Continual update	novel-unit minimum support	$\max(8, Q_{0.25})$
	merge threshold	$1.25\times$ existing radius
	micro-cluster radius	bootstrap NN-distance P95

Table 9: Baseline methods and their main discovery mechanisms.

Method	Discovery mechanism
AutoPlait	MDL-based joint segmentation and regime discovery
TICC	Toeplitz inverse-covariance clustering with temporal smoothness
Time2State	causal-CNN representation followed by a DPGMM
E2USD	frequency-aware representation followed by a DPGMM
CLaP	self-supervised classification and segment merging
CompILE	variational learning of segment boundaries and latent codes

learn segment representations (Kipf et al., 2019). Per-channel normalization is also a standard preprocessing practice in time-series analysis for reducing sensitivity to offsets and amplitude scales.

CLaP. The released implementation requires candidate change points as input rather than detecting them internally. We obtain these candidates using ClaSP, the change-point detector on which CLaP is based, and otherwise use the released implementation unchanged.

CompILE. The released implementation is designed for discrete symbolic sequences and therefore requires minor adaptation to our continuous multivariate streams. We replace its discrete embedding with a linear projection and use squared error instead of cross-entropy for reconstruction, while leaving its boundary inference and latent-variable model unchanged. Long recordings are divided into 2,000-frame sequences because CompILE uses a fixed segment budget per sequence and processing entire long recordings is impractical. Finally, we cluster the resulting segment-level latent representations with a single dataset-wide Dirichlet process mixture so that state identities are consistent across recordings, following the clustering protocol used by Time2State and E2USD.

All experiments ran on a single machine with GPUs (NVIDIA RTX A6000 and A40). Time2State, E2USD and CompILE use a GPU; AutoPlait, TICC and CLaP are CPU-bound.

E.3 Hyperparameter Selection

No hyperparameter is selected using evaluation performance. We follow the released configurations and dataset-specific settings of each baseline where available, and otherwise keep settings fixed across datasets.

Table 10: Hyperparameters used for the baseline methods. Values follow the released implementations or their reference experiments unless noted otherwise.

Method	Setting	Value
AutoPlait	parameter-free	–
TICC	number of states	8
	window size	1
	switch penalty	600
	sparsity	0.11
	max iterations	100
Time2State	encoder channels / depth	30 / 10
	reduced size / output channels	80 / 4 (9 on PAMAP2)
	kernel size / learning rate	3 / 3×10^{-3}
	M / N	20 / 4 (10 / 4 on MoCap)
	training steps	20 (40 on PAMAP2)
E2USD	encoder depth	1
	M / N	20 / 4
	output channels	4 (9 on PAMAP2)
	training steps	20
CLaP	window size	SuSS
	classifier	ROCKET
	merge score	classification gain
CompILE	segments per sequence	8
	latent / hidden dimension	32 / 64
	learning rate	10^{-3}
	training steps	100
	sequence length	2,000 frames

F Extra results

Table 11 reports mean ± standard deviation over five seeds for Time2State, E2USD, TICC and RONDO. TICC is deterministic for a given input, so its spread comes only from the seed-dependent PCA fit and is zero on MoCap and PAMAP2, which use no PCA. AutoPlait is deterministic and CLaP uses the fixed random state of its released implementation, so both are run once. CompILE is run once, with seed 0, because of its computational cost.

Table 11: Complete results (%), mean ± standard deviation over five seeds. For SegRatio, values closer to 1 are better.

Method	MoF	Recall	F1@50	Edit	SegRatio	Method	MoF	Recall	F1@50	Edit	SegRatio
MoCap						**Drive&Act**					
Time2State	88.68 ± 2.07	91.49 ± 2.36	84.48 ± 2.29	83.80 ± 1.17	1.70 ± 0.13	Time2State	54.23 ± 0.39	29.55 ± 1.79	7.00 ± 0.39	12.97 ± 0.47	11.31 ± 0.13
E2USD	84.97 ± 2.81	87.45 ± 6.11	77.95 ± 5.77	76.48 ± 6.78	1.97 ± 0.28	E2USD	47.60 ± 0.48	18.90 ± 0.81	5.66 ± 0.29	11.80 ± 0.19	14.47 ± 0.42
TICC	86.61 ± 0.00	84.52 ± 0.00	79.39 ± 0.00	83.18 ± 0.00	2.19 ± 0.00	TICC	49.47 ± 0.28	23.71 ± 1.39	6.61 ± 0.65	13.01 ± 0.70	9.14 ± 0.09
AutoPlait	90.27	90.28	91.18	92.05	0.97	AutoPlait	39.42	8.08	0.26	4.32	0.04
CompILE	66.45	61.05	52.19	63.14	1.97	CompILE	46.17	17.20	9.04	32.69	4.13
CLaP	74.61	65.63	60.53	62.39	0.74	CLaP	49.26	17.74	11.43	20.10	0.37
RONDO	88.91 ± 1.66	96.78 ± 1.91	95.06 ± 2.37	96.09 ± 2.86	2.42 ± 0.28	RONDO	68.36 ± 0.20	62.30 ± 2.42	14.29 ± 0.63	22.22 ± 0.36	8.04 ± 0.13
PAMAP2						**OPPORTUNITY**					
Time2State	79.69 ± 0.40	92.13 ± 1.87	6.17 ± 0.20	8.65 ± 0.14	24.72 ± 0.48	Time2State	69.28 ± 0.65	48.94 ± 1.07	20.80 ± 1.46	58.15 ± 1.08	2.08 ± 0.05
E2USD	76.58 ± 1.48	90.21 ± 2.21	4.22 ± 0.49	7.25 ± 0.43	26.96 ± 1.30	E2USD	69.02 ± 2.21	49.95 ± 2.77	26.49 ± 1.32	50.32 ± 2.20	2.91 ± 0.20
TICC	74.71 ± 0.00	78.69 ± 0.00	15.93 ± 0.00	16.42 ± 0.00	7.27 ± 0.00	TICC	65.91 ± 0.80	42.99 ± 1.01	13.66 ± 0.59	21.28 ± 1.29	0.37 ± 0.01
AutoPlait	45.22	29.81	21.46	21.45	0.97	AutoPlait	37.64	20.83	0.00	0.63	0.01
CompILE	34.11	8.28	4.65	10.41	15.92	CompILE	47.13	25.54	3.44	25.24	0.46
CLaP	77.40	79.70	50.00	51.30	3.03	CLaP	58.88	29.61	2.82	6.61	0.09
RONDO	84.12 ± 0.93	94.25 ± 1.66	22.91 ± 1.95	20.82 ± 0.80	15.37 ± 0.79	RONDO	79.96 ± 0.27	59.19 ± 0.31	30.59 ± 1.10	48.06 ± 1.73	1.23 ± 0.03
ActRecTut						**LARa**					
Time2State	73.40 ± 2.20	49.08 ± 2.11	42.20 ± 0.73	49.07 ± 2.19	1.10 ± 0.07	Time2State	69.91 ± 0.17	37.95 ± 0.75	27.09 ± 0.83	45.89 ± 0.67	2.00 ± 0.01
E2USD	72.87 ± 0.75	46.83 ± 2.44	40.41 ± 2.06	45.03 ± 4.89	1.01 ± 0.04	E2USD	67.24 ± 0.83	35.62 ± 2.08	21.81 ± 1.40	40.36 ± 1.39	2.13 ± 0.02
TICC	69.68 ± 0.36	41.59 ± 1.66	29.33 ± 1.02	32.24 ± 1.06	0.45 ± 0.03	TICC	70.40 ± 0.15	35.57 ± 0.69	28.34 ± 1.03	44.85 ± 0.95	1.92 ± 0.02
AutoPlait	44.42	14.06	2.04	2.31	0.01	AutoPlait	57.46	17.22	1.36	5.06	0.04
CompILE	45.69	27.96	8.76	31.11	0.44	CompILE	60.03	31.30	11.20	26.13	0.57
CLaP	51.68	27.70	5.45	5.45	0.02	CLaP	65.28	23.93	7.47	10.76	0.16
RONDO	75.66 ± 0.66	52.85 ± 3.34	36.71 ± 2.40	45.26 ± 1.15	0.77 ± 0.03	RONDO	78.44 ± 0.18	53.01 ± 0.78	38.50 ± 0.77	47.34 ± 0.51	1.31 ± 0.01