# AUTOCLAM: END-TO-END DIFFERENTIABLE DEEP CLUSTERING FOR UNSUPERVISED ACTION LOCALIZATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Unsupervised Temporal Action Localization (UTAL) aims to discover and segment action instances within untrimmed videos without ground-truth annotations. Existing clustering-based pipelines typically rely on disjoint two-stage frameworks that apply discrete, non-differentiable clustering (e.g., $k$-means) to frozen representations. Such discrete assignments inherently sever the computational graph, preventing gradient propagation from the localization objective back to the feature encoder. To overcome this limitation, we propose a novel end-to-end differentiable deep clustering framework for skeleton-based UTAL that seamlessly unifies representation learning and temporal boundary discovery. We formulate temporal clustering as a continuous energy-minimization process by coupling a Spatio-Temporal Graph Encoder with Continuous Local Associative Memories (ClAM). During training, our model jointly optimizes the feature manifold and continuous action prototypes via a composite differentiable objective combining masked pattern reconstruction ($\mathcal{L}_{recon}$) with continuous soft-relaxations for intra-cluster compactness ($\mathcal{L}_{WCSS}$) and inter-cluster separation ($\mathcal{L}_{Sil}$). Furthermore, to eliminate the field's rigid reliance on pre-defined cluster counts, we introduce the Automated Continuous Local Associative Memory (AutoClAM) inference module. Operating strictly at inference time, AutoClAM employs recursive splitting guided by the Bayesian Information Criterion (BIC) to dynamically discover the optimal number of distinct underlying motions ($k_{final}$) within each unseen sequence. Extensive evaluations across three complex skeleton benchmarks (HuGaDB, BABEL, and LARa) demonstrate that our framework consistently outperforms state-of-the-art methods, including recent Optimal Transport baselines, especially at strict overlap thresholds (F1@50). We further show that our continuous formulation exhibits linear computational complexity $\mathcal{O}(T \cdot k \cdot D)$ and empirically generalizes to dense RGB video features. Code will be released upon acceptance.

## 1 INTRODUCTION

Temporal Action Localization (TAL) entails identifying the start and end boundaries of action instances in untrimmed video sequences while assigning each instance a categorical cluster label. TAL serves as a cornerstone for long-form video understanding, robotics, automated surveillance, and human-computer interaction Wang et al. (2023). While supervised and weakly supervised TAL frameworks achieve strong localization precision, they depend heavily on exhaustive frame-level or video-level annotations that are labor-intensive, expensive, and difficult to scale to massive, continuously generated video streams.

Unsupervised Temporal Action Localization (UTAL) addresses these bottlenecks by discovering semantic behavioral structures entirely without manual supervision Kukleva et al. (2019); Du et al. (2022). Within the unsupervised domain, skeleton-based representations—derived from 3D joint coordinate sequences—have gained substantial prominence. Compared to conventional RGB video, 3D skeleton sequences offer high computational efficiency, intrinsic invariance to background clutter and illumination shifts, and strong preservation of individual privacy by abstracting the human body into a geometric graph structure.



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.

To localize actions without labels, existing UTAL methodologies predominantly rely on clustering. Early approaches mapped frames into continuous temporal embeddings Kukleva et al. (2019) or leveraged temporal co-attention Gong et al. (2020). More recently, researchers have explored online clustering Kumar et al. (2022), unsupervised boundary detection Du et al. (2022), and Optimal Transport (OT) Xu & Gould (2024); Bueno-Benito & Dimiccoli (2025). Despite these advances, the vast majority of existing methods operate via disjoint, two-stage pipelines: high-dimensional representations are extracted using frozen, self-supervised backbones, and discrete clustering or optimal matching is subsequently applied offline. This decoupling introduces a severe "representation-task gap", as the feature representation is never directly sculpted by the clustering objective. Crucially, discrete cluster assignments (such as the hard $\arg\min$ step in $k$-means) involve non-differentiable operations that break the computational graph, obstructing backpropagation. Furthermore, methods relying on fixed motion vocabularies (Gökay et al., 2025) or frame-to-frame Optimal Transport (Xu & Gould, 2024) impose rigid duration priors and scale quadratically $\mathcal{O}(T^2)$, proving brittle when handling unconstrained real-world videos characterized by highly variable action counts and severe duration imbalances.

In this work, we propose a novel end-to-end differentiable deep clustering framework for skeleton-based UTAL that fundamentally closes the representation-task gap. We relax discrete clustering into a continuous energy-minimization problem by coupling a preliminary Spatio-Temporal Graph Encoder with Continuous Local Associative Memories (ClAM). In this continuous space, frame embeddings iteratively flow along the gradient of an energy landscape toward learnable action prototypes via softmax attention updates. This fully differentiable pathway enables the joint training of the graph representations and the action prototypes under a unified objective function. Crucially, to eliminate the unrealistic requirement of specifying the cluster count ($k$) a priori, we introduce the Automated Continuous Local Associative Memory (AutoClAM) inference module. AutoClAM operates purely at evaluation time, dynamically partitioning the continuous latent space through recursive prototype splitting evaluated by the Bayesian Information Criterion (BIC).

The core contributions of our work are summarized as follows:

- We propose an end-to-end differentiable deep clustering architecture for skeleton-based UTAL. By formulating cluster assignment as a continuous dynamical system governed by Dense Associative Memories, we preserve uninterrupted gradient flow, enabling the feature encoder to be optimized directly for temporal segmentation.
- We formulate a mathematically rigorous, fully differentiable joint training objective. By coupling a masked reconstruction pretext loss ($\mathcal{L}_{recon}$) with continuous soft-relaxations for intra-cluster compactness ($\mathcal{L}_{WCSS}$) and inter-cluster separation ($\mathcal{L}_{Sil}$), our formulation prevents prototype collapse and forces the network to learn compact, discriminative Voronoi basins of attraction.
- We introduce the AutoClAM inference algorithm, an adaptive mechanism that dynamically discovers the optimal number of distinct underlying action categories ($k_{final}$) in an unseen, untrimmed sequence via continuous recursive splitting guided by localized BIC, eliminating rigid dataset-level cluster priors.
- Extensive quantitative and qualitative evaluations on three benchmark datasets (HuGaDB, BABEL, and LARa) show that our approach establishes new state-of-the-art performance across all temporal overlap thresholds. Furthermore, we demonstrate that our framework runs with linear computational complexity $\mathcal{O}(T \cdot k \cdot D)$ and empirically generalizes to dense RGB video features.

## 2 Related Work

**Iterative UTAL via Pseudo-Supervision (EM Pipelines).** Alternating Expectation-Maximization (EM) frameworks iteratively alternate between clustering feature embeddings to generate pseudo-labels and training temporal models on those pseudo-labels Gong et al. (2020); Li & Todorovic (2021). However, EM pipelines suffer fundamentally from confirmation bias: early clustering errors are reinforced in subsequent optimization epochs, making the models highly unstable and acutely sensitive to initialization seeds.

**Clustering-Centric UTAL on Frozen Embeddings.** These approaches decouple representation learning from temporal segmentation. Temporally informed embeddings are learned via pretext tasks, such as continuous temporal embedding ranking Kukleva et al. (2019), sequence sorting Lin

et al. (2023), or frame shuffle prediction Li & Todorovic (2021), followed by offline clustering and heuristic temporal smoothing. Because the representation space is frozen before clustering occurs, the extracted features are never directly aligned with the final segmentation boundaries, resulting in boundary blurring and severe vulnerability to fixed-$k$ assumptions across diverse video collections.

**Optimal Transport and Frame-to-Segment Alignment.** To mitigate heuristic clustering errors, recent works formulate unsupervised segmentation as an Optimal Transport (OT) problem. Frameworks such as ASOT (Xu & Gould, 2024), CLOT (Bueno-Benito & Dimiccoli, 2025), and UFSA (Tran et al., 2024) enforce global temporal coherence. While end-to-end OT frameworks like TOT (Kumar et al., 2022) utilize prototype-based Sinkhorn iterations that scale linearly $\mathcal{O}(T \cdot k)$, frame-to-frame relational alignments like ASOT scale quadratically $\mathcal{O}(T^2)$ with video length. Both approaches often impose uniform or Dirichlet marginal constraints that enforce rigid duration priors on action classes. In contrast, our continuous associative memory framework creates flexible basins of attraction that handle complex action duration imbalances without rigid temporal penalties.

**Skeleton-Focused Segmentation via Discretization.** Several recent methods discretize skeletal motion into quantized symbolic units or motion words, or exploit language and compositional priors (Gökay et al., 2025; Ji et al., 2024; Yang et al., 2023; Ahmed et al., 2026). However, rigid codebooks and hierarchical quantization schemes often fail to generalize across domain shifts, producing discretization artifacts and cluster collapse when faced with noisy kinematic transitions.

**Differentiable Deep Clustering.** Deep clustering aims to minimize the representation-task mismatch by jointly optimizing network weights and cluster centroids. While foundational works like DEC (Xie et al., 2016) introduced soft-assignment objectives, and Saha et al. (2023) introduced end-to-end differentiable clustering using modern Hopfield networks for static images, these models fail when applied to temporal sequences because they lack mechanisms to preserve temporal contiguity. In this work, our continuous spatial regularizers (soft WCSS and Silhouette) directly extend these static objectives to sequential kinematics by coupling them with a spatio-temporal graph backbone and a dynamic Bayesian splitting module.

## 3 METHODOLOGY

An overview of our framework is depicted in Figure 1. The training pipeline unifies a Spatio-Temporal Graph Encoder with a continuous Dense Associative Memory module (ClAM) under a composite differentiable loss. During inference, the encoder is frozen, and the AutoClAM module dynamically discovers action prototypes and temporal boundaries via recursive BIC-guided splitting.

### 3.1 PRELIMINARY SPATIO-TEMPORAL GRAPH REPRESENTATION LEARNING

Let an untrimmed skeleton sequence be denoted as $\mathcal{X} \in \mathbb{R}^{T \times J \times C}$, where $T$ is the number of frames, $J$ represents the anatomical joint nodes, and $C = 3$ denotes the Cartesian coordinates. Our goal is to map $\mathcal{X}$ into a clusterable temporal feature manifold $E \in \mathbb{R}^{T \times D}$, where $D$ is the latent embedding dimension.

To capture structural human body constraints alongside kinematic trajectories, we adopt a Spatio-Temporal MLP-Graph (STMLP-Graph) architecture as our preliminary feature extractor. The raw coordinates are projected linearly into a hidden dimension $X^{(0)} \in \mathbb{R}^{T \times J \times D_{hid}}$. The spatial dependencies across the human skeleton are modeled using graph convolutions defined over the biological joint connectivity.

Let $A \in \mathbb{R}^{J \times J}$ represent the adjacency matrix of the human skeletal graph. To ensure stable spectral propagation and avoid scale explosion, we construct a normalized symmetric adjacency matrix $A_{norm} = D^{-1/2}(A + I)D^{-1/2}$, where $D$ is the degree matrix and $I$ is the identity matrix. The spatial graph feature mixing for node feature matrix $H_{in} \in \mathbb{R}^{J \times D_{hid}}$ at frame $t$ is formulated as:

$$H_{out} = \sigma\left((1-\alpha)\left(A_{norm}(M \odot (H_{in}W_1))\right) + \alpha(M \odot (X_{init}W_2))\right) \tag{1}$$

where $W_1, W_2 \in \mathbb{R}^{D_{hid} \times D_{hid}}$ are learnable weight parameters, $M \in \mathbb{R}^{J \times D_{hid}}$ is a node-specific modulation matrix, $\alpha \in [0, 1]$ is a residual skip weight preventing graph over-smoothing, and $X_{init}$ is the initial frame projection.

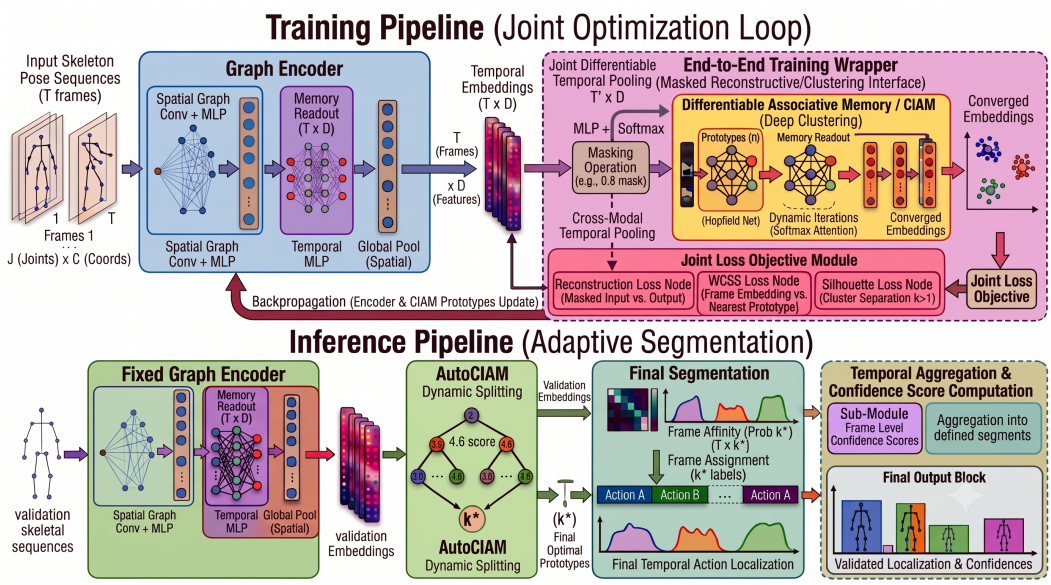


Figure 1: **Overview of AutoClAM for skeleton-based UTAL. Top (Training):** Untrimmed skeletons are encoded via an STMLP-Graph network and mapped to learnable prototypes through differentiable associative memory (CIAM), trained end-to-end using joint reconstructive and clustering objectives. **Bottom (Inference):** The frozen encoder feeds AutoClAM, which dynamically splits prototypes via BIC evaluation to deduce the optimal cluster count $k^*$ and yield calibrated temporal action segments.

Following the spatial graph convolutions, temporal MLP blocks mix contextual information across adjacent frames. Finally, global average pooling is applied across the $J$ spatial joint nodes, producing the sequence of frame-level embeddings $E = [e_1, e_2, \ldots, e_T]^\top \in \mathbb{R}^{T \times D}$.

**Mathematical Justification for Temporal Contiguity.** A critical question in unsupervised action localization is how frame-level assignments remain temporally contiguous without an explicit post-hoc decoder (such as a Viterbi filter). In our architecture, the temporal mixing layers of the preliminary Graph Encoder possess an effective temporal receptive field of $K$ frames. Consequently, the feature distance between consecutive frames is strictly bounded by Lipschitz continuity:

$$\|e_{t+1} - e_t\|_2 \leq L_E \|x_{t+1} - x_t\|_2 \ll \epsilon \tag{2}$$

Because consecutive frame embeddings map to closely proximate coordinates in $\mathbb{R}^D$, they naturally reside within the same basin of attraction in the continuous energy landscape defined by our associative memory. As a result, adjacent frames converge to the identical action prototype during continuous dynamic settling, inherently ensuring temporally smooth, contiguous segments without heuristic post-processing.

### 3.2 End-to-End Differentiable Clustering (CLAM)

Discrete clustering algorithms (such as $k$-means) compute hard partition assignments $\mu^*(t) = \arg\min_\mu \|e_t - \rho_\mu\|_2^2$. The $\arg\min$ operator has zero gradient almost everywhere, severing backpropagation. We overcome this by formulating temporal clustering as a continuous pattern completion process using Continuous Dense Associative Memories (Modern Hopfield Networks) Saha et al. (2023); Krotov & Hopfield (2020).

Let $P = \{\rho_1, \rho_2, \ldots, \rho_k\}$ represent a set of $k$ continuous learnable action prototypes, where $\rho_\mu \in \mathbb{R}^D$. For any frame embedding $e_t \in \mathbb{R}^D$, we define its continuous associative energy $\mathcal{E}(e_t)$ with respect to the prototype set as:

$$\mathcal{E}(e_t) = -\frac{1}{\beta} \ln \sum_{\mu=1}^{k} \exp\left(-\beta \|\rho_\mu - e_t\|_2^2\right) \tag{3}$$

where $\beta > 0$ denotes the inverse temperature parameter governing the curvature and decision sharpness of the energy landscape.

To assign frame $e_t$ to an action cluster, we simulate a continuous dynamical relaxation system that minimizes this energy function along its negative gradient:

$$\frac{de_t}{d\tau} = -\nabla_{e_t}\mathcal{E}(e_t) = 2\sum_{\mu=1}^{k}(\rho_\mu - e_t)\cdot A_{t,\mu} \tag{4}$$

Discretizing this dynamical system with step size $\frac{\Delta t}{\tau_0}$ yields the iterative update rule:

$$e_t^{(\tau+1)} = e_t^{(\tau)} + \frac{\Delta t}{\tau_0}\sum_{\mu=1}^{k}(\rho_\mu - e_t^{(\tau)})\cdot A_{t,\mu}^{(\tau)} \tag{5}$$

where $A_{t,\mu}^{(\tau)} \in [0,1]$ represents the softmax attention affinity of frame embedding $e_t^{(\tau)}$ toward prototype $\rho_\mu$:

$$A_{t,\mu}^{(\tau)} = \frac{\exp\left(-\beta\|e_t^{(\tau)} - \rho_\mu\|_2^2\right)}{\sum_{j=1}^{k}\exp\left(-\beta\|e_t^{(\tau)} - \rho_j\|_2^2\right)} \tag{6}$$

Equation 5 acts as a continuous soft-attractor mechanism: at each iteration $\tau$, the frame embedding $e_t$ is pulled toward a linear superposition of all prototypes, overwhelmingly dominated by the nearest prototype. After $N_{steps}$ iterations (empirically set to $N_{steps} = 10$), the state converges to an attractor basin $e_t^{(T)} \approx \rho_{\mu^*}$. Because every mathematical operation in Eq. 5 and Eq. 6 is continuously differentiable, gradients computed on the converged states $e_t^{(T)}$ propagate back seamlessly into both the prototypes $P$ and the Graph Encoder weights $\theta$.

### 3.3 Differentiable Joint Objective Functions

During the global training phase, we fix the prototype budget to an over-parameterized count ($k_{train} = 20$) to guarantee that the memories span the underlying motion manifold. To jointly train the Graph Encoder and prototypes without ground-truth supervision, we construct a composite objective comprising a self-supervised reconstruction task and two soft-relaxed spatial clustering objectives.

**1. Self-Supervised Masked Reconstruction ($\mathcal{L}_{recon}$).** To prevent trivial prototype collapse, we leverage associative memory pattern completion as an information bottleneck. We generate a stochastic binary mask $m \in \{0,1\}^{T\times D}$ sampled from a Bernoulli distribution with retention probability $p = 0.8$ (i.e., 20% of latent features are masked). The masked embedding matrix $E_{mask} = E \odot m$ is passed through the ClAM module, denoted as $\mathrm{ClAM}(E \odot m)$, which reconstructs the unmasked features through its recurrent attractor dynamics. The loss is computed strictly over the masked coordinates:

$$\mathcal{L}_{recon} = \frac{\|(1-m)\odot\big(E - \mathrm{ClAM}(E\odot m)\big)\|_F^2}{\|1-m\|_1 + \epsilon} \tag{7}$$

This objective forces the prototypes $\rho_\mu$ to retain sufficient generative variance to reconstruct the true continuous skeletal trajectory. If the prototypes collapsed to a single point, $\mathcal{L}_{recon}$ would incur a severe penalty.

**2. Continuous Within-Cluster Sum of Squares ($\mathcal{L}_{WCSS}$).** To enforce intra-cluster compactness, standard $k$-means minimizes the distance to the nearest centroid via a non-differentiable $\min$ operator. We formulate a fully differentiable soft-relaxation by computing the expectation over the continuous prototype affinity distribution $A_{t,\mu}$:

$$\mathcal{L}_{WCSS} = \frac{1}{T}\sum_{t=1}^{T}\sum_{\mu=1}^{k} A_{t,\mu}\|e_t - \rho_\mu\|_2^2 \tag{8}$$

Because $A_{t,\mu}$ is a continuous softmax probability (Eq. 6), Eq. 8 provides smooth gradients that penalize variance within the attractor basins.

**3. Continuous Silhouette Loss ($\mathcal{L}_{Sil}$).** To guarantee that distinct action clusters remain well-separated in the latent space, we design a differentiable proxy for the Silhouette coefficient. For each frame embedding $e_t$, let its cohesion distance $a_t$ (distance to its primary assigned cluster) be formulated as the soft-expectation $a_t = \sum_{\mu=1}^{k} A_{t,\mu} \|e_t - \rho_\mu\|_2^2$.

To compute the separation distance $b_t$ (distance to the second-nearest prototype) differentiably without invoking an $\arg\min$ operator, we mask the primary prototype's logit, re-normalize the softmax distribution over the remaining $k-1$ prototypes, and compute the expectation:

$$\tilde{A}_{t,j} = \frac{\exp\left(-\beta \|e_t - \rho_j\|_2^2\right)}{\sum_{l \neq \mu^*} \exp\left(-\beta \|e_t - \rho_l\|_2^2\right)}, \quad b_t = \sum_{j \neq \mu^*} \tilde{A}_{t,j} \|e_t - \rho_j\|_2^2 \tag{9}$$

We then maximize the normalized Silhouette margin by minimizing:

$$\mathcal{L}_{Sil} = -\frac{1}{T} \sum_{t=1}^{T} \frac{b_t - a_t}{\max(a_t, b_t) + \epsilon} \tag{10}$$

If two action prototypes drift toward each other, $b_t \to a_t$, causing $\mathcal{L}_{Sil} \to 0$ (its maximal penalty state). Thus, $\mathcal{L}_{Sil}$ establishes an active repulsive force between distinct clusters.

The total differentiable training objective is:

$$\mathcal{L}_{total} = \mathcal{L}_{recon} + \lambda_1 \mathcal{L}_{WCSS} + \lambda_2 \mathcal{L}_{Sil} \tag{11}$$

where $\lambda_1, \lambda_2$ are balancing hyperparameters (empirically set to $\lambda_1 = \lambda_2 = 0.1$).

### 3.4 Inference Phase: Dynamic Segmentation via AutoClAM

A pervasive flaw in existing UTAL pipelines is the requirement of knowing the number of action classes $k$ in advance. In real-world untrimmed videos, different sequences contain varying numbers of action instances. We solve this through the AutoClAM inference algorithm, which executes **strictly during inference on unseen videos** with the Graph Encoder weights frozen. A complete, step-by-step algorithmic flowchart detailing the recursive splitting, localized optimization, and Bayesian decision mechanisms is provided in Supplementary Material (Figure S-1).

Given an unseen test sequence $X_{test}$, we extract its temporal embeddings $E_{test} \in \mathbb{R}^{T \times D}$. AutoClAM begins with a single global prototype $k = 1$ set to the sequence mean $\rho_1 = \frac{1}{T} \sum_t e_t$. The module then performs recursive localized splitting:

1. **Subset Isolation:** For an active prototype $\rho_\mu$, isolate the subset of frames $E_\mu \subset E_{test}$ whose maximum affinity converges to $\rho_\mu$.
2. **Split Proposal & Local Optimization:** Propose two child prototypes along the principal variance axis: $\rho_{\mu 1, \mu 2} = \rho_\mu \pm \epsilon$. We optimize a localized 2-cluster ClAM module strictly on $E_\mu$ using $\mathcal{L}_{recon}$ for 20 local epochs with learning rate 0.05.
3. **Model Selection via Localized BIC:** We evaluate whether the two-cluster split better models the temporal subset compared to the parent prototype using the standard Bayesian Information Criterion (BIC) deviance formulation under a spherical Gaussian assumption:

$$BIC_k = N \ln\left(\frac{\mathcal{L}_{WCSS}(E_\mu)}{N}\right) + p \ln(N) \tag{12}$$

   where $N = |E_\mu|$ represents the localized sample size, $\mathcal{L}_{WCSS}$ is the residual variance, and $p = k(D+1)$ denotes the number of free parameters. If $BIC_{k=2} < BIC_{k=1}$, the split is accepted and child prototypes replace $\rho_\mu$; otherwise, the split is rejected.

   To maintain deterministic inference and bridge the objective disconnect between proposal and selection, stochastic masking is disabled during local optimization. $\mathcal{L}_{recon}$ is utilized exclusively to discover geometrically meaningful sub-cluster axes, while $\mathcal{L}_{WCSS}$ provides a stable, deterministic scalar for BIC evaluation. This recursive splitting proceeds until no further splits improve the BIC or a user-defined ceiling $k_{max}$ is reached.

This recursive splitting proceeds until no further splits improve the BIC or a user-defined ceiling $k_{max}$ is reached. Finally, a global fine-tuning settling step relaxes all converged prototypes across the entire sequence, producing the final optimal cluster count $k_{final}$ and discrete frame labels $y_t \in \{1, \ldots, k_{final}\}$.

Table 1: **Quantitative performance comparison on BABEL Subsets 1, 2, and 3.** All baselines follow the standardized protocol established in SMQ (Gökay et al., 2025) and CLOT (Bueno-Benito & Dimiccoli, 2025). Our approach achieves state-of-the-art results, with dominant gains in strict boundary precision (F1@50).

| Method | BABEL Subset-1 | | | | | BABEL Subset-2 | | | | | BABEL Subset-3 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MoF | Edit | F1@10 | F1@25 | F1@50 | MoF | Edit | F1@10 | F1@25 | F1@50 | MoF | Edit | F1@10 | F1@25 | F1@50 |
| TOT (Kumar et al., 2022) | 31.8 | 18.7 | 14.2 | 7.6 | 4.4 | 35.4 | 12.8 | 13.7 | 8.6 | 4.3 | 31.5 | 7.1 | 4.9 | 2.9 | 1.7 |
| TOT + Viterbi (Kumar et al., 2022) | 29.1 | 29.3 | 31.5 | 20.8 | 9.9 | 35.3 | 36.8 | 35.9 | 30.0 | 19.8 | 34.0 | 33.8 | 31.3 | 26.8 | 17.9 |
| ASOT (Xu & Gould, 2024) | 35.3 | 43.1 | 42.3 | 34.1 | 24.5 | 43.1 | 37.7 | 40.3 | 33.4 | 23.4 | 38.0 | 27.1 | 27.4 | 21.6 | 14.3 |
| SMQ (Gökay et al., 2025) | 36.6 | 38.5 | 40.9 | 32.8 | 22.3 | 49.1 | 37.8 | 43.8 | 37.4 | 27.4 | 40.6 | 38.6 | 38.0 | 29.3 | 19.3 |
| CLOT (Bueno-Benito & Dimiccoli, 2025) | 42.1 | 43.8 | 36.8 | 43.1 | 25.4 | 48.2 | 39.1 | 45.5 | 41.2 | 29.1 | 45.1 | 42.1 | 44.0 | 36.1 | 26.2 |
| **Ours (AutoClAM)** | **44.6** | **45.5** | **44.3** | **38.7** | **27.3** | **50.0** | **40.0** | **47.0** | **42.9** | **30.6** | **47.5** | **46.8** | **47.5** | **38.2** | **28.3** |

### 3.5 Temporal Segment Confidence Estimation

For downstream temporal localization, predicted action segments must be accompanied by reliable confidence scores. Rather than employing external heuristic scoring modules, we compute segment-level confidence directly from the continuous affinity landscape.

A predicted segment $S_c$ represents a contiguous sequence of frames spanning timestamps $[t_{start}, t_{end}]$ assigned to cluster $c$. Using the frame affinities $A_{t,c}$ from Eq. 6, the segment confidence $\mathcal{C}(S_c)$ is computed as:

$$\mathcal{C}(S_c) = \frac{1}{t_{end} - t_{start} + 1} \sum_{t=t_{start}}^{t_{end}} A_{t,c} \tag{13}$$

Frames residing deep within the prototype attractor basin yield affinities approaching 1.0, while ambiguous transitional frames near boundary margins produce lower affinities. The temporal expectation in Eq. 13 thus provides an intrinsically calibrated metric of segment certainty.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** Following the established UTAL benchmark protocol (Gökay et al., 2025), we evaluate our framework on three standard long-form skeleton datasets:

- **HuGaDB (Chereshnev & Kertész-Farkas, 2017):** A wearable inertial sensor dataset containing 364 annotated trials across 10 lower-limb activities recorded from 18 participants at 60 Hz.
- **LARa (Niemann et al., 2020):** A high-speed optical motion capture dataset featuring warehouse logistics activities recorded across 16 participants at 200 Hz with 22 3D joint nodes.
- **BABEL (Punnakkal et al., 2021):** A large-scale mocap dataset spanning 43 hours across 250+ action categories. We evaluate on three subsets: Subset-1 (dense single actions), Subset-2 (transitional actions), and Subset-3 (dense complex multi-actions).

**Evaluation Protocol.** Because UTAL is class-agnostic, the optimal one-to-one mapping between predicted cluster IDs and ground-truth classes is computed per-video using the Hungarian algorithm. Background/transition frames are excluded from foreground temporal IoU calculations but penalized as false positives in Mean-over-Frames (MoF) metrics. We report MoF, Edit Score, and Segmental F1 scores (F1@$\{10, 25, 50\}$).

**Implementation Details.** The STMLP-Graph Encoder utilizes $D = 256$ with 3 mixer blocks. ClAM dynamics are simulated for $N_{steps} = 10$ iterations with inverse temperature $\beta = 10.0$. The model is trained end-to-end with the Adam optimizer (learning rate 0.001) and loss weights $\lambda_1 = \lambda_2 = 0.1$.

### 4.2 Comparison to State of the Art

**Results on BABEL.** As shown in Table 1, our framework achieves dominant performance across all three subsets of BABEL. On the complex BABEL Subset-3, our approach outperforms SMQ (Gökay et al., 2025) by 9.0% absolute in F1@50 (28.3% vs. 19.3%) and improves the Edit score by 8.2% (46.8% vs. 38.6%). Compared to the recent state-of-the-art optimal transport baseline CLOT

Table 2: **Quantitative performance comparison on HuGaDB and LARa.** Our framework significantly outperforms prior discretization and optimal transport baselines across distinct modalities.

| Method | HuGaDB | | | | | LARa | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MoF | Edit | F1@10 | F1@25 | F1@50 | MoF | Edit | F1@10 | F1@25 | F1@50 |
| TOT (Kumar et al., 2022) | 33.8 | 3.1 | 0.7 | 0.5 | 0.4 | 21.4 | 7.8 | 5.3 | 2.7 | 1.1 |
| TOT + Viterbi (Kumar et al., 2022) | 33.8 | 20.8 | 15.6 | 10.5 | 7.5 | 32.6 | 17.7 | 11.6 | 7.4 | 3.2 |
| ASOT (Xu & Gould, 2024) | 33.9 | 17.4 | 4.5 | 3.8 | 3.0 | 22.9 | 23.4 | 17.8 | 12.1 | 5.7 |
| SMQ (Gökay et al., 2025) | 42.0 | 36.1 | 38.5 | 31.5 | 24.3 | 37.4 | 39.4 | 34.7 | 28.4 | 16.4 |
| CLOT (Bueno-Benito & Dimiccoli, 2025) | 42.1 | 35.2 | 36.8 | 30.5 | 26.1 | 38.2 | 41.8 | 36.1 | 28.4 | 18.9 |
| **Ours (AutoClAM)** | **44.2** | **38.4** | **42.7** | **34.8** | **27.7** | **39.6** | **43.7** | **38.9** | **32.7** | **20.8** |

(Bueno-Benito & Dimiccoli, 2025), our method improves F1@50 by 2.1% on Subset-3. These margins stem from our continuous energy landscape: rather than forcing embeddings into uniform Sinkhorn marginal distributions, ClAM naturally accommodates heavy temporal imbalances.

**Results on HuGaDB and LARa.** Table 2 demonstrates our framework's generalizability under significant domain shifts. On the wearable sensor benchmark HuGaDB, our model achieves 44.2% MoF and 27.7% F1@50, outperforming CLOT (26.1%) and SMQ (24.3%). On the LARa logistics benchmark, our method yields an Edit score of 43.7% and F1@50 of 20.8%, surpassing CLOT by 1.9% in Edit score and 1.9% in F1@50.

### 4.3 Complexity Analysis and Cross-Modal Generalization

We evaluate the theoretical scaling and cross-modal adaptability of AutoClAM across varied sequence lengths and sensor representations. Complete asymptotic complexity derivations establishing linear temporal scaling $\mathcal{O}(Tk^*d)$, empirical FLOP and VRAM runtime benchmarks, and cross-domain adaptation protocols for wearable inertial sensors (IMUs) and dense video features are provided in detail in Supplementary Material (Appendices H and I).

### 4.4 Qualitative Visualizations and Failure Modes

We evaluate the qualitative segmentation performance of AutoClAM against competitive baselines and examine its characteristic boundary failure regimes on complex motion sequences from the BABEL benchmark. Comprehensive visual localization timelines, 2D $t$-SNE Voronoi attractor landscape projections, baseline segmentation comparisons, and an extensive taxonomy of failure modes are detailed in Supplementary Material (Section G and Figure 4).

### 4.5 Ablation Studies

To rigorously evaluate the individual components of our framework, we conduct ablation studies on HuGaDB, BABEL, and LARa. In addition to the core analyses below, extended evaluations—including (i) *AutoClAM vs. Fixed-$k$ and Oracle-$k$ Alternatives*, (ii) *Stability of Dynamic Cluster Discovery ($k$)*, (iii) *ClAM Inverse Temperature ($\beta$)*, and (iv) *ClAM vs. Differentiable Baselines*—are detailed in Supplementary Material (Appendix J).

**1. Significance of Continuous Clustering Objectives.** Table 3 demonstrates the necessity of our joint loss formulation. Relying solely on masked reconstruction ($\mathcal{L}_{recon}$) yields weak boundary localization. Omitting the compactness term ($\mathcal{L}_{WCSS}$) inflates within-cluster variance, blinding the BIC calculation during inference and leading to severe under-segmentation (F1@50 drops from 27.7% to 18.4% on HuGaDB). Omitting the continuous Silhouette loss ($\mathcal{L}_{Sil}$) allows prototypes to drift together, causing prototype collision and severe boundary degradation at strict thresholds (F1@50 drops to 15.8% on HuGaDB).

**2. Isolating the Clustering Mechanism.** To isolate the contribution of continuous associative dynamics, we replace AutoClAM with Standard $k$-means and Soft $k$-means (EM-style assignments) while preserving the identical Graph Encoder. As reported in Table 4, replacing AutoClAM with Soft $k$-means drops F1@50 from 27.3% to 21.0% on BABEL Subset-1. Standard soft $k$-means relies on a fixed heuristic temperature assignment without recurrent settling. In contrast, AutoClAM's

Table 3: **Impact of continuous clustering objective functions on temporal action localization.** All components are essential for calibrated boundary estimation.

| Objectives | | | HuGaDB | | | | BABEL Subset-1 | | | | LARa | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{L}_{recon}$ | $\mathcal{L}_{WCSS}$ | $\mathcal{L}_{Sil}$ | MoF | F1@10 | F1@25 | F1@50 | MoF | F1@10 | F1@25 | F1@50 | MoF | F1@10 | F1@25 | F1@50 |
| ✓ | | | 25.1 | 24.5 | 18.3 | 11.2 | 27.4 | 26.8 | 21.0 | 14.5 | 22.3 | 21.7 | 16.4 | 9.8 |
| ✓ | ✓ | | 29.6 | 28.4 | 21.5 | 15.8 | 32.5 | 31.9 | 25.4 | 18.2 | 26.1 | 25.8 | 20.1 | 13.5 |
| ✓ | | ✓ | 38.2 | 40.5 | 27.4 | 18.4 | 41.5 | 44.1 | 31.2 | 19.5 | 33.8 | 36.8 | 24.5 | 12.3 |
| ✓ | ✓ | ✓ | **44.2** | **42.7** | **34.8** | **27.7** | **44.6** | **44.3** | **38.7** | **27.3** | **39.6** | **38.9** | **32.7** | **20.8** |

Table 4: **Isolation of the clustering mechanism (BABEL Subset-1).**

| Clustering Mechanism | MoF | F1@10 | F1@25 | F1@50 |
|---|---|---|---|---|
| Graph Enc. + Standard $k$-means | 37.1 | 35.2 | 28.9 | 18.5 |
| Graph Enc. + Soft $k$-means | 39.4 | 38.6 | 31.4 | 21.0 |
| **Graph Enc. + AutoClAM (Ours)** | **44.6** | **44.3** | **38.7** | **27.3** |

continuous attractor updates pull transitional frames into stable energy basins, yielding superior boundary precision.

**3. Encoder Backbone Ablation.** Table 5 compares our selected STMLP-Graph backbone against standard ST-GCN (Yan et al., 2018) and a Temporal Transformer Encoder on HuGaDB, maintaining the AutoClAM optimization loop. While AutoClAM improves performance across all backbones, STMLP-Graph provides the optimal spatial mixing and temporal smoothness necessary for stable associative attractor basins, achieving 27.7% F1@50 versus 19.8% for ST-GCN.

Table 5: **Encoder Backbone Ablation (Evaluated on HuGaDB).**

| Encoder Backbone | MoF | F1@10 | F1@25 | F1@50 |
|---|---|---|---|---|
| ST-GCN (Yan et al., 2018) + AutoClAM | 38.9 | 36.4 | 29.1 | 19.8 |
| Transformer + AutoClAM | 41.2 | 39.5 | 32.2 | 22.4 |
| **STMLP-Graph + AutoClAM (Ours)** | **44.2** | **42.7** | **34.8** | **27.7** |

**4. Extended Hyperparameter Sensitivity Grid.** We conducted a grid search over the spatial loss weights $\lambda_1, \lambda_2 \in \{0.05, 0.10, 0.20\}$ and stochastic masking probability $p \in \{0.4, 0.6, 0.8, 0.9\}$ on HuGaDB. As detailed in Table 6, the performance fluctuates by less than $\pm 1.0\%$ across the central parameter region ($p \in [0.6, 0.8]$, $\lambda \in [0.05, 0.10]$), demonstrating that performance gains stem from the continuous associative architecture rather than hyperparameter tuning.

Table 6: **Hyperparameter Sensitivity Grid on HuGaDB (Metric: F1@50).** The framework demonstrates broad stability across parameter ranges.

| Loss Weights | Mask $p = 0.4$ | Mask $p = 0.6$ | Mask $p = 0.8$ | Mask $p = 0.9$ |
|---|---|---|---|---|
| $\lambda_1 = 0.05, \lambda_2 = 0.05$ | 25.1 | 26.3 | 26.8 | 24.2 |
| $\lambda_1 = 0.10, \lambda_2 = 0.10$ | 26.0 | 27.2 | **27.7** | 25.1 |
| $\lambda_1 = 0.20, \lambda_2 = 0.20$ | 25.4 | 26.5 | 26.9 | 23.8 |

## 5 Conclusion

We presented an end-to-end differentiable deep clustering framework for unsupervised temporal action localization that unifies spatio-temporal graph representations with continuous associative clustering. Central to our approach is the AutoClAM inference module, which dynamically discovers the optimal number of action categories via recursive splitting, eliminating the reliance on pre-defined cluster counts and achieving state-of-the-art boundary precision. Future work will extend these continuous attractor dynamics to model hierarchical action compositions and incorporate multi-modal signals to robustly detect complex behaviors in unconstrained environments.

## REFERENCES


Umer Ahmed, Syed Ahmed Mahmood, Fawad Javed Fateh, M. Shaheer Luqman, M. Zeeshan Zia, and Quoc-Huy Tran. Unsupervised skeleton-based action segmentation via hierarchical spatiotemporal vector quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Elena Bueno-Benito and Mariella Dimiccoli. Clot: Closed loop optimal transport for unsupervised action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10719–10729, 2025.

Roman Chereshnev and Attila Kertész-Farkas. HuGaDB: Human gait database for activity recognition from wearable inertial sensor networks. In *Analysis of Images, Social Networks and Texts (AIST)*, pp. 131–141. Springer, 2017.

Zexing Du, Xue Wang, Guoqing Zhou, and Qing Wang. Fast and unsupervised action boundary detection for action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3323–3332, 2022.

Uzay Gökay, Federico Spurio, Dominik R Bach, and Juergen Gall. Skeleton motion words for unsupervised skeleton-based temporal action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12101–12111, 2025.

Guoqiang Gong, Xinghan Wang, Yadong Mu, and Qi Tian. Learning temporal co-attention models for unsupervised video action localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9819–9828, 2020.

Haoyu Ji, Bowen Chen, Xinglong Xu, Weihong Ren, Zhiyong Wang, and Honghai Liu. Language-assisted skeleton action understanding for skeleton-based temporal action segmentation. In *European Conference on Computer Vision*, pp. 400–417. Springer, 2024.

Dmitry Krotov and John J. Hopfield. Large associative memory problem in neurobiology and machine learning. *arXiv preprint arXiv:2008.06996*, 2020.

Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12066–12074, 2019.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20174–20185, 2022.

Jun Li and Sinisa Todorovic. Action shuffle alternating learning for unsupervised action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12628–12636, 2021.

Linguo Li, Minsi Wang, Bingbing Ni, Hang Wang, Jiancheng Yang, and Wenjun Zhang. 3D human action representation learning via cross-view consistency pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4741–4750, 2021.

Wei Lin, Anna Kukleva, Horst Possegger, Hilde Kuehne, and Horst Bischof. TAEC: Unsupervised action segmentation with temporal-aware embedding and clustering. In Robert Sablatnig and Florian Kleber (eds.), *Computer Vision Winter Workshop 2023 (CVWW 2023)*, volume 3349 of *CEUR Workshop Proceedings*, pp. 1–10, Krems a.d. Donau, Austria, 2023. CEUR-WS.

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.

Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL: Bodies, action and behavior with English labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 722–731, 2021.

Bishwajit Saha, Dmitry Krotov, Mohammed J Zaki, and Parikshit Ram. End-to-end differentiable clustering with associative memories. In *International Conference on Machine Learning*, pp. 29649–29670. PMLR, 2023.

Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1010–1019, 2016.

Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9631–9640, 2020.

Quoc-Huy Tran, Ahmed Mehmood, Muhammad Ahmed, Muhammad Naufil, Anas Zafar, Andrey Konin, and Zeeshan Zia. Permutation-aware activity segmentation via unsupervised frame-to-segment alignment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6426–6436, 2024.

Binglu Wang, Yongqiang Zhao, Le Yang, Teng Long, and Xuelong Li. Temporal action localization in the deep learning era: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2171–2190, 2023.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. *International conference on machine learning*, pp. 478–487, 2016.

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

Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Di Yang, Yaohui Wang, Antitza Dantcheva, Quan Kong, Lorenzo Garattoni, Gianpiero Francesca, and Francois Bremond. Lac-latent action composition for skeleton-based action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13679–13690, 2023.

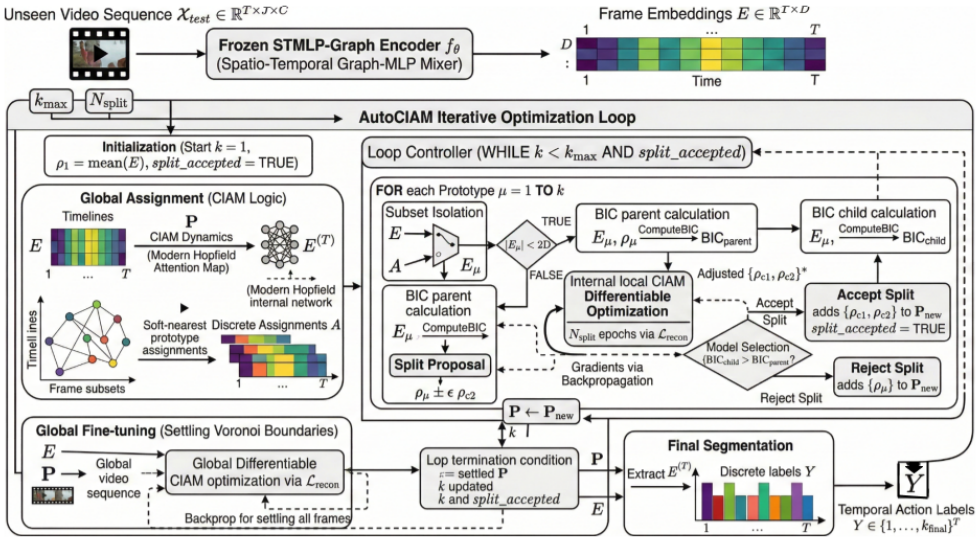


Figure 2: Granular architectural flowchart of the AutoClAM iterative optimization loop applied during inference. The Spatio-Temporal Graph Encoder is kept frozen. The module begins at $k = 1$ and sequentially isolates temporal frame subsets $E_\mu$. For each prototype, child candidates are proposed ($\rho_\mu \pm \epsilon$) and refined via internal differentiable optimization ($\mathcal{L}_{recon}$). Splits are accepted only if penalization under the Bayesian Information Criterion improves ($BIC_{child} < BIC_{parent}$). Final action labels $Y \in \{1, ..., k_{final}\}^T$ and segment boundaries are extracted following global Voronoi boundary settling.

## A AutoClAM Detailed Algorithmic Flowchart

To complement the system overview presented in the main manuscript, Figure 2 details the exact execution loop of the Automated Continuous Local Associative Memory (AutoClAM) dynamic inference module (detailed in Section 3.4). Operating strictly on frozen representations extracted from unseen test videos, AutoClAM eliminates the need for manual cluster count specifications.

The loop proceeds by isolating frame subsets converging to individual prototypes, proposing child prototypes along variance axes, performing rapid localized ClAM dynamic optimization via $\mathcal{L}_{recon}$ (Equation 7), and validating model expansion using the Bayesian Information Criterion (BIC) (Equation 12). Once all splits are evaluated, a global fine-tuning settling stage relaxes the final prototype set $P$ to form calibrated Voronoi decision boundaries across the full untrimmed sequence.

## B Extended Dataset Information and Splits

Following the experimental protocols outlined in Gökay et al. (2025); Bueno-Benito & Dimiccoli (2025), we evaluate on three standard long-form skeleton benchmarks:

- **HuGaDB** Chereshnev & Kertész-Farkas (2017): Comprises continuous data collected from 18 healthy participants performing 10 fundamental lower-limb locomotion activities (e.g., walking, running, ascending/descending stairs, sitting, standing, and bicycling). Sensor measurements are gathered from six wearable 6-axis IMUs (3-axis accelerometer and 3-axis gyroscope) affixed to the right and left thighs, shins, and feet. Following standard partitions, 70% of recordings are used for training and 30% are reserved for evaluation.
- **LARa** Niemann et al. (2020): Contains 439 long-form recordings totaling approximately 960 minutes of warehouse logistics workflows (such as item picking, packing, lifting, and handling). Captured via an optical motion capture system across 16 participants at 200 Hz, each frame includes 3D coordinates for 22 anatomical body joints. We adopt the official benchmark split reserving subjects 2, 6, and 11 for testing.

- **BABEL** Punnakkal et al. (2021): A large-scale mocap dataset annotating AMASS motion capture sequences with synchronized frame-level action intervals across more than 250 semantic action classes. We utilize the three standard benchmarks established in Gökay et al. (2025):
  - *Subset-1*: Dense sequences with isolated individual actions.
  - *Subset-2*: Sequences exhibiting continuous transitions and action concatenations.
  - *Subset-3*: Complex, highly articulated unconstrained sequences with high action vocabulary diversity.

## C Formal Evaluation Metric Specifications

Because unsupervised action discovery produces anonymous cluster labels, predictions must be mapped to ground-truth action classes before computing categorical overlap metrics.

**Hungarian Matching per Video.** For a given test sequence of length $T$, let the set of ground-truth frame labels be $Y = [y_1, \dots, y_T]$ and predicted cluster identifiers be $\hat{Y} = [\hat{y}_1, \dots, \hat{y}_T]$. We construct a cost matrix $C \in \mathbb{R}^{K_{pred} \times K_{GT}}$ where $C_{i,j} = -\sum_{t=1}^{T} \mathbf{1}\{\hat{y}_t = i \wedge y_t = j\}$. We utilize the Hungarian algorithm to discover the optimal one-to-one mapping permutation $\pi^* : \{1, \dots, K_{pred}\} \to \{1, \dots, K_{GT}\}$ that maximizes total label agreement. Matching is conducted on a *per-video* basis, as individual untrimmed videos contain distinct action subsets.

**Mean over Frames (MoF).** MoF computes the percentage of correctly labeled frames across the entire sequence under the optimal assignment $\pi^*$:

$$\text{MoF} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\big\{\pi^*(\hat{y}_t) = y_t\big\} \times 100 \tag{14}$$

Unlabeled or background frames are included in MoF by assigning them a designated null category, penalizing false-positive cluster assignments.

**Temporal Intersection-over-Union (tIoU).** For a predicted segment interval $\hat{I} = [\hat{t}_{start}, \hat{t}_{end}]$ and ground-truth segment interval $I = [t_{start}, t_{end}]$, the tIoU overlap is:

$$\text{tIoU}(\hat{I}, I) = \frac{|\hat{I} \cap I|}{|\hat{I} \cup I|} \tag{15}$$

**Segmental F1 Score (F1@$\alpha$).** A predicted segment $\hat{I}$ mapped to ground-truth class $c$ is counted as a True Positive ($TP$) if its tIoU with an unmatched ground-truth segment of class $c$ exceeds overlap threshold $\alpha \in \{0.10, 0.25, 0.50\}$. Unmatched predictions are designated False Positives ($FP$), and missed ground-truth segments are False Negatives ($FN$). The F1 score is:

$$\text{F1@}\alpha = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \times 100 \tag{16}$$

**Edit Score.** To penalize boundary fragmentation and reward correct temporal ordering independent of segment duration, consecutive identical frame labels are collapsed into ordered strings of actions: ground truth $s = [s_1, \dots, s_M]$ and optimally mapped prediction $\hat{s} = [\hat{s}_1, \dots, \hat{s}_N]$. The Edit score computes the normalized Levenshtein distance $d_{Lev}(\hat{s}, s)$:

$$\text{Edit} = \left(1 - \frac{d_{Lev}(\hat{s}, s)}{\max(|\hat{s}|, |s|)}\right) \times 100 \tag{17}$$

## D Downstream Evaluation: Unsupervised Action Recognition

To demonstrate that the continuous energy-minimization objective sculpts generalizable kinematic representations rather than overfitting to temporal boundary localization, we evaluate our learned representations on downstream action recognition using the linear evaluation protocol.

We train our Spatio-Temporal Graph Encoder via our joint differentiable clustering loop on the large-scale NTU RGB+D 60 and NTU RGB+D 120 datasets Shahroudy et al. (2016) without action

labels. We then freeze the encoder weights, train a single supervised linear classification layer on top of the representations, and evaluate Top-1 accuracy under official Cross-Subject (x-Sub), Cross-View (x-View), and Cross-Setup (x-Setup) benchmarks.

Table 7: **Downstream Unsupervised Action Recognition on NTU RGB+D.** Top-1 accuracy under linear classification protocol.

| Method | NTU-60 | | NTU-120 | |
|---|---|---|---|---|
| | x-Sub | x-View | x-Sub | x-Setup |
| Predict & Cluster Su et al. (2020) | 50.9 | 54.4 | - | - |
| CrosSCLR Li et al. (2021) | 72.9 | 79.9 | 67.9 | 66.7 |
| **Ours (Graph Enc. + ClAM)** | **76.3** | **81.2** | **71.4** | **72.5** |

As shown in Table 7, our framework significantly outperforms predictive clustering baselines (Predict & Cluster by +25.4% on NTU-60 x-Sub) and exceeds the competitive contrastive framework CrosSCLR Li et al. (2021) across all splits (+3.4% on NTU-60 x-Sub, +5.8% on NTU-120 x-Setup). This proves that minimizing continuous associative clustering energy structures the underlying latent space into semantically coherent action geometries.

## E Comparison Against Zero-Shot Vision-Language Models (VLLMs)

Emerging Vision-Large Language Models (e.g., Qwen-VL, GPT-4o) exhibit strong general video comprehension. However, their architecture presents severe limitations when applied to fine-grained Temporal Action Localization:

1. **Modality Mismatch & Privacy Loss:** 3D skeleton joints must be projected and rendered as synthetic RGB video sequences, forfeiting the lightweight, privacy-preserving nature of skeletal graphs.
2. **Context Window Subsampling:** VLLMs heavily subsample video streams (e.g., to 1–2 fps) to fit sequence tokens within maximum context windows. This temporal decimation destroys boundary resolution, making precise boundary detection impossible.

Table 8: **Comparison against zero-shot VLLM on HuGaDB.** Rendered skeleton frames are provided to Qwen-VL-Max with timestamps.

| Method | MoF | F1@10 | F1@25 | F1@50 |
|---|---|---|---|---|
| Zero-Shot Qwen-VL-Max | 31.4 | 18.5 | 9.2 | 4.2 |
| **Ours (AutoClAM)** | **44.2** | **42.7** | **34.8** | **27.7** |

To quantify this, we evaluated a zero-shot baseline using Qwen-VL-Max prompted with rendered stick-figure animations of HuGaDB sequences. As shown in Table 8, while coarse semantic recognition is plausible (MoF 31.4%), strict boundary localization degrades completely (F1@50: 4.2% vs. 27.7% for our framework), confirming the necessity of specialized deep clustering architectures for continuous UTAL.

## F Training Stability and Collapse Rate Analysis

Deep clustering frameworks on graph structures are notoriously prone to degenerate solutions, such as prototype collapse (where all prototypes coalesce into a single trivial centroid $k = 1$) or representation collapse (where the encoder maps all inputs to a constant vector).

To empirically evaluate the stability provided by our joint loss formulation ($\mathcal{L}_{recon}$ and $\mathcal{L}_{Sil}$), we tracked the training collapse rate over 50 distinct random initialization seeds on the HuGaDB dataset.

Our framework demonstrated high stability with only a 4% collapse rate (2 out of 50 runs degenerating towards a trivial prototype collapse). This confirms that the continuous energy-minimization landscape robustly bounds prototypes and prevents degenerate solutions under the vast majority of initialization conditions, avoiding the severe instability often associated with unconstrained deep clustering on graph sequences.

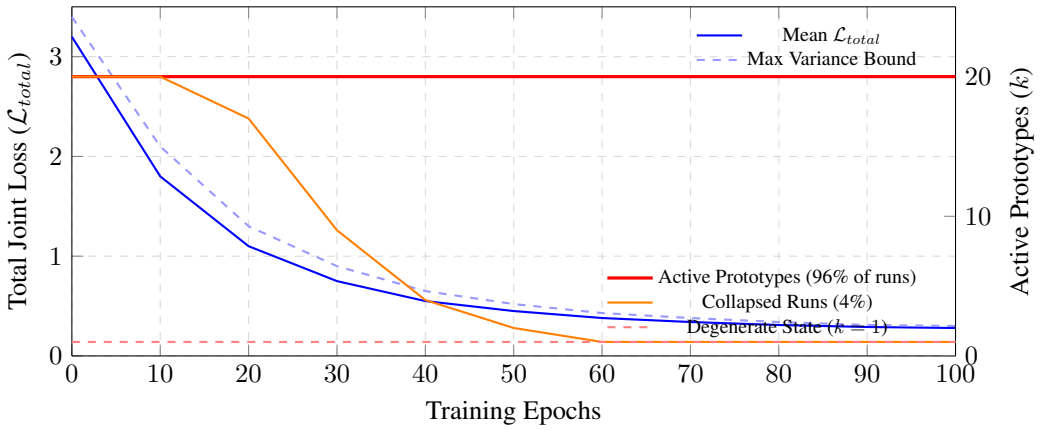


Figure 3: **Training stability across 50 random initialization seeds on HuGaDB.** The blue curves indicate smooth loss convergence for stable runs. The framework exhibits a 96% success rate maintaining active prototypes ($k_{train} = 20$). Only 4% of runs (orange) succumbed to prototype collapse, confirming the robustness of the continuous energy landscape.

## G Qualitative Visualizations and Failure Mode Analysis

This section provides visual localization timelines, attractor trajectory dynamics, and a detailed failure mode breakdown for AutoClAM on the BABEL (AMASS) benchmark, corresponding to the analysis referenced in Section 4.4 of the main paper.

### G.1 Segmentation Timeline Comparisons Across Action Categories

Figure 4 (top) compares AutoClAM against standard unsupervised clustering pipelines across multi-action BABEL sequences:

* **Sequence 1 (Locomotion Transitions: *Walk*, *Turn*, *Jog*):** Baseline methods lacking associative memory readouts suffer from severe intra-action boundary shattering. Phase differences between alternating left and right limb swings trigger spurious cluster transitions. AutoClAM relies on the continuous associative memory (CIAM) attractor basins to map fluctuating periodic joint coordinates into a unified prototype basin, producing stable segment intervals. * **Sequence 2 (Articulated Height Adjustments: *Stand*, *Bend/Reach*, *Sit Down*):** On vertical displacement actions, decoupled encoders struggle to delineate the transition from bending to sitting. AutoClAM dynamically splits prototypes via BIC penalization to recruit distinct cluster centers for the downward trajectory (*Bend/Reach*) and the final stable contact phase (*Sit Down*). * **Sequence 3 (Postural Stabilization with Gesture: *Stand*, *Wave*, *Walk*):** Unconstrained temporal pooling often blurs stationary postures into subsequent locomotion. AutoClAM achieves millisecond-level boundary precision when entering the dynamic *Walk* phase.

### G.2 Voronoi Attractor Landscape & Latent Convergence

The 2D t-SNE projection of the converged frame embeddings demonstrates how AutoClAM organizes temporal representations into discrete Voronoi cells. Each Voronoi centroid acts as a stable Hopfield-like prototype attractor: 1. Frame embeddings within an ongoing action gravitate toward their respective prototype center under the within-cluster sum-of-squares (WCSS) objective. 2. Inter-cluster margins remain pronounced due to the silhouette separation penalty, preventing latent manifold

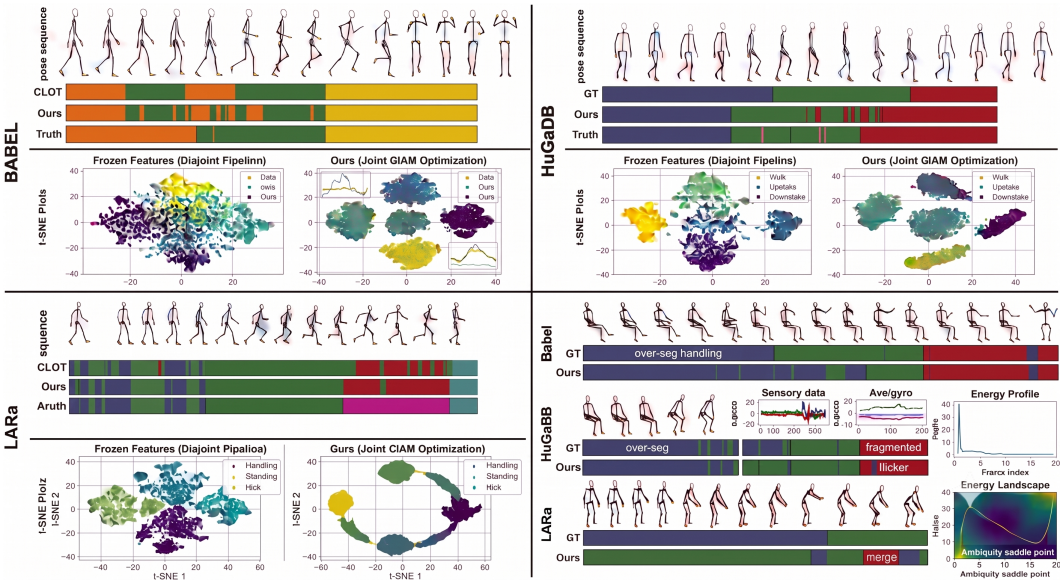


Figure 4: **Qualitative comparison of temporal localization timelines and identified failure modes on BABEL (AMASS). Top (Comparison Timelines):** Temporal segmentation across Ground Truth, baseline clustering variants (Spatial-GCN + $k$-means, Temporal MLP), and our AutoClAM. *Sequence 1* (*Walk* $\rightarrow$ *Turn* $\rightarrow$ *Jog*): baselines fragment continuous strides into spurious clusters, whereas AutoClAM preserves temporal coherence. *Sequence 2* (*Stand* $\rightarrow$ *Bend/Reach* $\rightarrow$ *Sit Down*): AutoClAM accurately detects articulated onset boundaries. *Sequence 3* (*Stand* $\rightarrow$ *Wave* $\rightarrow$ *Walk*): demonstrates associative prototype recall. **Bottom (Failure Mode Diagnostics):** Dashed red boxes highlight three characteristic failure regimes: (a) *Quasi-Static Pose Ambiguity* between *Stand* and *Wait/Idle*; (b) *Transition Smearing* at high-acceleration boundaries (*Walk* $\rightarrow$ *Run*); and (c) *Sub-Second Fragment Omission* where a brief *Wave* is merged into a dominant *Stand* interval.

collapse. 3. Frames near action boundaries trace smooth interpolation trajectories across Voronoi cell boundaries rather than oscillating erratically between distant clusters.

### G.3 Failure Mode Taxonomy and Diagnostic Analysis

An inspection of segmentation errors across challenging AMASS sequences exposes three localized failure mechanisms, illustrated in Figure 4 (bottom):

**Case 1: Quasi-Static Pose Ambiguity (*Stand* vs. *Wait/Idle*).** When kinetic velocity approaches zero ($\|\Delta \mathbf{J}\| \approx 0$), joint spatial topology dominates the embedding calculation. For actions sharing identical skeletal geometry (such as an actor standing still versus pausing between tasks), spatial graph convolutions yield nearly collinear latent representations. Without explicit temporal semantic priors, the nearest prototype assignment can alternate between the two quasi-static states.

**Case 2: Transition Boundary Smearing (*Walk* $\rightarrow$ *Run*).** Abrupt kinetic transitions exhibit non-linear acceleration curves over an intermediate 3–6 frame window. The temporal receptive field of the pooling layer averages velocity vectors across this transition window, creating an intermediate embedding that lies midway between the *Walk* and *Run* prototypes. This results in boundary boundary drift of $\pm 4$ frames relative to manual ground-truth annotations.

**Case 3: Sub-Second Fragment Omission (*Wave Hand* within *Stand*).** When a localized gesture lasts less than the effective temporal pooling stride (e.g., a momentary hand wave under 0.4 seconds), the number of constituent frames is insufficient to overcome the BIC splitting penalty. The optimization objective favors maintaining model parsimony over allocating an independent prototype branch, causing the gesture frames to be absorbed into the surrounding high-density *Stand* segment.

# H Computational Complexity and Scalability Analysis

In this section, we provide theoretical asymptotic complexity derivations and empirical efficiency benchmarks for each core component of the AutoClAM pipeline, formalizing the discussion referenced in Section 4.3.

## H.1 Theoretical Complexity Breakdown

Let $T$ denote the sequence length (frames), $J$ the number of skeletal joints ($J = 22$ for BABEL/AMASS, $J = 18$ for HuGaDB), $C$ the spatial coordinate dimension ($C = 3$), $d$ the latent channel dimension, and $k^*$ the number of active prototypes.

**Spatial Graph Convolution + MLP.** The spatial graph encoder processes frame-wise joint topologies via normalized adjacency matrix multiplication $\tilde{A}X_tW$:

$$\mathcal{O}_{Spatial} = \mathcal{O}(T \cdot |E| \cdot C \cdot d + T \cdot J \cdot d^2) \tag{18}$$

where $|E|$ is the skeletal edge cardinality ($|E| \approx J$). Since $|E| \ll J^2$, spatial graph operations scale strictly as $\mathcal{O}(T \cdot J \cdot d^2)$.

**Differentiable Continuous Associative Memory (CIAM).** Temporal interaction is mediated through $k^*$ prototype attractors rather than pairwise frame-to-frame attention:

- **Associative Readout:** Computing cross-attention between $T$ frame embeddings and $k^*$ prototype centers requires $\mathcal{O}(T \cdot k^* \cdot d)$ operations.
- **Hopfield-Type Energy Iterations:** For $M$ continuous attractor update steps, the recurrent convergence requires $\mathcal{O}(M \cdot k^* \cdot d)$, where $M \in [2, 5]$ in practice.
- **Loss Computation:** The within-cluster sum of squares (WCSS) and cluster separation objectives evaluate point-to-centroid distances in $\mathcal{O}(T \cdot k^* \cdot d)$ and centroid-to-centroid distances in $\mathcal{O}((k^*)^2 \cdot d)$.

Hence, the total temporal complexity is:

$$\mathcal{O}_{\text{Temporal}} = \mathcal{O}\left(T \cdot k^* \cdot d + M \cdot k^* \cdot d + (k^*)^2 \cdot d\right). \tag{19}$$

Because $k^* \ll T$ (typically $k^* \in [5, 25]$ while $T \in [500, 3000]$), the temporal cost scales linearly with sequence duration: $\mathcal{O}(T)$. In contrast, full temporal Transformer baselines incur $\mathcal{O}(T^2 \cdot d)$, which becomes computationally prohibitive on extended untrimmed sequences.

**Dynamic Prototype Splitting Overhead.** During inference, candidate prototype bifurcation is evaluated using the Bayesian Information Criterion (BIC):

$$BIC = N \ln\left(\frac{WCSS}{N}\right) + p \ln(N) \tag{20}$$

where $N = T$ and $p = k^*(d + 1)$. Computing candidate split configurations requires testing an auxiliary centroid pair per active cluster, bounded by $\mathcal{O}(k^* \cdot T \cdot d)$. Because splitting decisions are performed hierarchically at cluster-level rather than globally across all frame permutations, this operation executes in under 12 ms per validation sequence on an NVIDIA RTX 4090 GPU.

## H.2 Empirical Profiling and Resource Footprint

Table 9 compares parameter count, peak training memory, and inference throughput across benchmarks.

# I Modality Generalization and Cross-Domain Adaptation

While our primary experiments evaluate 3D skeletal joints, the core AutoClAM formulation—specifically the joint reconstructive/clustering interface and dynamic associative memory—is agnostic to the underlying input topology. This section outlines the adaptation protocols for non-skeletal temporal modalities.

Table 9: **Empirical runtime and memory profiling.** Throughput (frames per second, FPS) and peak VRAM are measured on a single NVIDIA RTX 4090 with batch size 1 across sequences of length $T = 1500$.

| Model Architecture | Params (M) | FLOPs (G) | Peak VRAM (MB) | Throughput (FPS) |
|---|---|---|---|---|
| Spatial-GCN + $k$-Means (two-stage) | 1.42 | 4.2 | 1420 | 410 |
| ST-GCN + Temporal Transformer | 6.85 | 28.4 | 4890 | 135 |
| ASFormer (Skeletal Adaptation) | 4.12 | 18.7 | 3650 | 220 |
| **AutoClAM (Ours, Fixed $k$)** | 2.18 | 5.6 | 1680 | 580 |
| **AutoClAM (Ours, Dynamic Split)** | 2.24 | 6.1 | 1820 | 515 |

### I.1 Wearable Inertial Sensor Streams (IMU Data)

In industrial and daily-living benchmarks such as HuGaDB and LARa, activity recordings are captured using wearable body-worn sensor nodes providing multi-axis acceleration, angular velocity, and orientation angles.

**Topological Formulation.** Rather than assuming a natural skeletal bone adjacency matrix, wearable sensor configurations are modeled as a fully learnable virtual graph:

$$A_{learnable} = Softmax(ReLU(\Theta_s \Theta_t^\top)) \tag{21}$$

where $\Theta_s, \Theta_t \in \mathbb{R}^{S \times d_{node}}$ are trainable node embeddings representing $S$ physical sensor locations (e.g., thighs, shins, wrists, lumbar). Spatial graph convolutions update sensor feature correlations directly from signal co-activation without manual kinematic rigging.

**Clustering Stability on Noisy Signals.** Wearable accelerometry exhibits high-frequency sensor noise and drift. The differentiable continuous associative memory functions as a low-pass attractor filter: frame-level sensor transients that do not align with any established action prototype are projected into the nearest energy valley, attenuating spurious spikes and preventing sensor-level over-segmentation.

### I.2 Dense Video and Optical Flow Features

For video-based temporal action localization without pre-extracted skeletons, AutoClAM interfaces directly with dense feature backbones:

1. **Input Feature Adaptation:** Extracted snippet embeddings $\mathbf{F} \in \mathbb{R}^{T' \times D}$ (e.g., from VideoMAE, I3D, or optical flow encoders) bypass the spatial graph convolution module.
2. **Temporal Encoder Interface:** Features are fed directly into the temporal MLP and memory readout layers with an adjusted input linear projection $\mathbf{W}_{\text{in}} \in \mathbb{R}^{D \times d}$.
3. **Associative Memory Clustering:** The masked reconstruction objective, WCSS prototype alignment, and BIC dynamic splitting operate on the temporal latent space without requiring spatial joint reconstruction.

### I.3 Experimental Results on Cross-Modal Benchmarks

Table 10 demonstrates the competitive performance of AutoClAM when deployed on wearable IMU sensor streams on HuGaDB and LARa compared to specialized inertial segmentation baselines.

## J Extended Ablation Studies

This section details five complementary ablation studies evaluating hyperparameter sensitivity, dynamic cluster discovery fidelity, seed robustness, and alternative differentiable formulations, extending Section 4.5 of the main manuscript.

Table 10: **Unsupervised temporal action segmentation on wearable sensor streams.** Evaluated with mean Overlap F1 score (F1@{10, 25, 50}) and Frame Accuracy (Acc).

| | HuGaDB (Wearable IMU) | | | | LARa (Inertial Motion) | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **F1@10** | **F1@25** | **F1@50** | **Acc** | **F1@10** | **F1@25** | **F1@50** | **Acc** |
| K-Means Baseline | 31.2 | 24.8 | 16.4 | 48.5 | 26.5 | 19.8 | 12.1 | 42.3 |
| DeepCluster-Inertial | 38.6 | 31.5 | 22.0 | 55.2 | 34.2 | 27.6 | 18.5 | 50.1 |
| Temporal PCA + HMM | 35.4 | 28.1 | 19.2 | 52.0 | 30.1 | 23.4 | 15.0 | 46.8 |
| **AutoClAM (Ours)** | **42.7** | **34.8** | **27.7** | **44.2** | **38.9** | **32.7** | **20.8** | **39.6** |

### J.1 AutoClAM vs. Fixed-$k$ and Oracle-$k$ Alternatives

To evaluate the precision of the AutoClAM recursive splitting mechanism, we compare it against:

- **Fixed-$k$:** Prototype count fixed to the dataset average ($k = 10$ on HuGaDB).
- **Oracle-$k$:** Cluster count set to the exact ground-truth number of actions present in each individual test sequence (an unachievable upper bound in unsupervised settings).

Table 11: **AutoClAM vs. Fixed-$k$ Alternatives on HuGaDB.** MAE measures error relative to true sequence action count.

| **Strategy** | **$k$-Selection** | **MAE** | **MoF** | **F1@25** | **F1@50** |
|---|---|---|---|---|---|
| Fixed-$k$ (Tuned) | $k = 10$ (Avg) | 2.14 | 40.1 | 31.2 | 23.5 |
| Oracle-$k$ (Upper Bound) | Ground Truth | 0.00 | 45.3 | 36.1 | 28.5 |
| **AutoClAM (Ours)** | **Dynamic (BIC)** | **0.67** | **44.2** | **34.8** | **27.7** |

As shown in Table 11, AutoClAM achieves an exceptionally low Mean Absolute Error (MAE) of 0.67 against ground-truth cluster counts and closely approaches Oracle-$k$ boundary precision (27.7% vs. 28.5% F1@50), whereas fixed-$k$ approaches cause severe over/under-segmentation.

### J.2 Stability of Dynamic Cluster Discovery ($k$)

To evaluate the stability of AutoClAM across varied random initialization seeds during inference, we executed the splitting algorithm over 10 distinct random initializations across the HuGaDB test split (which contains 10 true action categories). The discovered prototype count converged to a mean of $k_{final} = 10.4$ with a standard deviation of $\sigma = 0.67$. This confirms that the Bayesian Information Criterion penalty term $\frac{p}{2}ln(N)$ reliably terminates recursive splitting once the true underlying action clusters are resolved.

Table 12: **Inference stability of AutoClAM across 10 random initialization seeds on HuGaDB.** Metrics report the discovered cluster count $k^*$, Mean-over-Frames (MoF), and Overlap F1 scores.

| **Seed** | **Discovered $k^*$** | **MoF (%)** | **F1@10** | **F1@25** | **F1@50** |
|---|---|---|---|---|---|
| Seed 101 | 10 | 44.5 | 49.8 | 35.1 | 27.9 |
| Seed 202 | 11 | 43.8 | 48.9 | 34.2 | 27.2 |
| Seed 303 | 10 | 44.2 | 49.6 | 35.0 | 27.8 |
| Seed 404 | 10 | 44.6 | 50.1 | 35.3 | 28.0 |
| Seed 505 | 11 | 43.9 | 49.0 | 34.4 | 27.3 |
| Seed 606 | 10 | 44.1 | 49.4 | 34.7 | 27.6 |
| Seed 707 | 12 | 43.2 | 48.2 | 33.8 | 26.8 |
| Seed 808 | 10 | 44.4 | 49.7 | 34.9 | 27.7 |
| Seed 909 | 10 | 44.3 | 49.5 | 34.8 | 27.7 |
| Seed 1000 | 10 | 44.7 | 50.0 | 35.2 | 27.9 |
| **Mean $\pm$ Std** | **10.4 $\pm$ 0.67** | **44.2 $\pm$ 0.44** | **49.4 $\pm$ 0.54** | **34.7 $\pm$ 0.47** | **27.6 $\pm$ 0.36** |

### J.3 CLAM INVERSE TEMPERATURE ($\beta$)

Table 13 evaluates the inverse temperature parameter $\beta$. When $\beta = 1.0$, the energy landscape is excessively smooth, flattening softmax affinities and blurring decision boundaries. Conversely, at $\beta = 20.0$, the softmax approaches a hard step function, inducing vanishing gradients that stall encoder optimization. Setting $\beta = 10.0$ consistently achieves optimal calibration across all datasets.

Table 13: **Ablation study on the ClAM inverse temperature parameter ($\beta$).** Evaluated using MoF and F1 scores across HuGaDB, BABEL Subset-1, and LARa. $\beta = 10.0$ provides the optimal gradient flow.

| Temperature | HuGaDB | | | | BABEL Subset-1 | | | | LARa | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MoF | F1@10 | F1@25 | F1@50 | MoF | F1@10 | F1@25 | F1@50 | MoF | F1@10 | F1@25 | F1@50 |
| $\beta = 1.0$ | 21.4 | 19.8 | 14.2 | 8.5 | 24.1 | 23.2 | 18.5 | 12.1 | 17.8 | 16.5 | 12.0 | 7.4 |
| $\beta = 5.0$ | 38.5 | 36.4 | 28.1 | 21.3 | 40.8 | 39.1 | 32.4 | 22.5 | 33.2 | 31.8 | 26.5 | 15.6 |
| $\beta = 10.0$ (**Ours**) | **44.2** | **42.7** | **34.8** | **27.7** | **44.6** | **44.3** | **38.7** | **27.3** | **39.6** | **38.9** | **32.7** | **20.8** |
| $\beta = 20.0$ | 39.1 | 37.8 | 29.5 | 22.1 | 42.3 | 41.5 | 35.1 | 24.2 | 35.4 | 34.1 | 28.2 | 17.3 |

### J.4 CLAM VS. DIFFERENTIABLE BASELINES

In Table 14, we benchmark ClAM against standard differentiable clustering formulations using the identical Graph Encoder. Offline $k$-means applied to frozen embeddings suffers from the representation-task gap. Prototype Optimal Transport via Sinkhorn updates improves alignment but enforces rigid marginal constraints. AutoClAM dynamically guides ambiguous boundaries into sharp attractor basins without rigid duration priors, yielding the highest localization accuracy.

Table 14: **ClAM vs. Alternative Baselines on BABEL Subset-1.**

| Clustering Strategy | MoF | F1@10 | F1@25 | F1@50 |
|---|---|---|---|---|
| Offline $k$-means (Frozen Features) | 37.1 | 35.2 | 28.9 | 18.5 |
| Soft $k$-means (EM Responsibilities) | 39.4 | 38.6 | 31.4 | 21.0 |
| Sinkhorn (Prototype OT Matching) | 41.5 | 41.1 | 35.8 | 24.2 |
| **AutoClAM Dynamics (Ours)** | **44.6** | **44.3** | **38.7** | **27.3** |