Learning Probabilistic Embeddings for Unsupervised Action Segmentation

Shuai Li[1], Duc Manh Vu[1], and Juergen Gall[1,2]

[1] University of Bonn, Germany
[2] Lamarr Institute for Machine Learning and Artificial Intelligence, Germany
{lishuai,gall}@iai.uni-bonn.de

Abstract. This paper concerns the problem of unsupervised temporal action segmentation for long, untrimmed videos. Recent successful approaches follow a joint representation learning and clustering paradigm, where optimal transport (OT) is adopted to produce pseudo labels for learning frame representations. These approaches alternate between estimating pseudo labels using OT and optimizing the parameters with gradient descent during training, where OT is used for obtaining the final temporal action segmentation. A major limitation of these works is that they learn a deterministic embedding for frame representations. The iterative procedure between learning deterministic embeddings based on pseudo labels and estimating pseudo labels from the learned embedding can thus get quickly stuck in a local optimum. As an alternative, we thus propose to learn a probabilistic embedding for frame representations. The embeddings are modeled by Gaussian distributions and we sample from the distributions before estimating the pseudo labels. We evaluate our approach on several challenging temporal action segmentation datasets and achieve results comparable to, and in some cases, better than the state of the art. Compared to baselines with deterministic embeddings, our approach improves MoF up to 20.7% and F1-score up to 19.0%. Our code is available at https://github.com/derkbreeze/PEOT.

Keywords: Unsupervised Temporal Action Segmentation · Probabilistic Embedding · Optimal Transport

1 Introduction

Unsupervised temporal action segmentation is highly relevant for many applications, such as monitoring and optimizing workflows in manufacturing, phenotyping human or animal behavior, as well as human-robot interaction and collaboration [23, 39]. The task, however, is challenging, as videos can contain different numbers of actions and actions can happen in a different order within the video. Furthermore, the same action can occur multiple times within a video. To tackle this problem, approaches based on joint representation learning and clustering [17, 32] have recently become popular. They iterate during training between estimating pseudo-labels, using optimal transport (OT), and updating the frame embeddings by using the estimated pseudo-labels as target for the cross-entropy

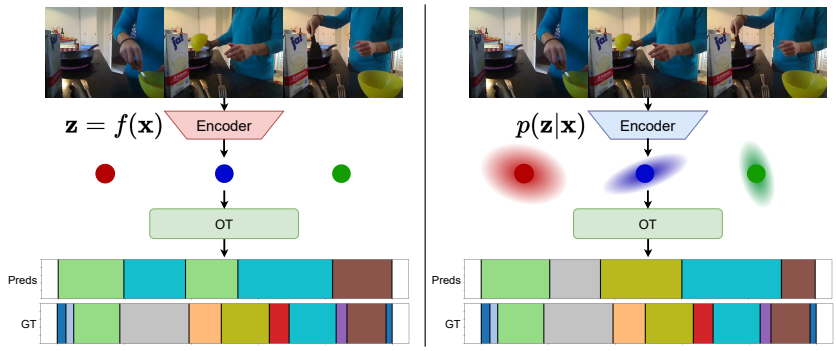


Fig. 1: Most previous works learn deterministic embeddings as frame embeddings; we propose to learn probabilistic embeddings such that frame representations are samples from Gaussian distributions that explicitly model embedding uncertainty. Our representations lead to more accurate segmentations compared to the baseline [38]. Different colors indicate different actions.

loss. Following this paradigm, Xu and Gould [38] recently proposed ASOT. The core idea is to use a combination of Kantorovic and Gromov-Wasserstein optimal transport, which offers temporal consistency. They also relax the balanced action assignment assumption in prior works [17, 32] via an unbalanced OT formulation. Similar to the idea of [32], CLOT [3] extends ASOT by building a three-level OT that introduces feedback between frame embeddings and action embeddings, which improves the detection of short segments.

We notice that all previous unsupervised works [3, 16–18, 30, 32, 34] learn a deterministic embedding as frame representations before computing pseudo-labels. This, however, has the disadvantage that the optimization using optimal transport can get very quickly stuck in a local optimum such that the deterministic embedding overfits to the wrong pseudo-labels. In this work, we thus propose to learn probabilistic frame embeddings using Gaussian distributions as illustrated in Fig. 1. We then sample features for each frame from the learned Gaussian distributions, and apply OT to compute the pseudo labels on the sampled features. For estimating the probabilistic frame embeddings, we find that Graph Convolutional Networks (GCN) [13] perform better than an MLP, as it is commonly used for deterministic frame embeddings, or a TCN [7].

We evaluate our approach on four challenging unsupervised temporal action segmentation benchmarks, namely Breakfast [14], Youtube Instructional [1], 50Salads [28], and Desktop Assembly [17], where the results are comparable to, and in some cases, better than the state of the art. More importantly, we demonstrate that the probabilistic frame embeddings improve unsupervised temporal action segmentation compared to a deterministic embedding, using state-of-the-art approaches like ASOT [38] or VASOT [2] as baselines. Compared to ASOT, our approach improves MoF up to 20.7% (+12.2) and F1-score up to 19.0% (+12.1). Compared to VASOT, our approach improves MoF up to 16.5% (+8.8)

and F1-score up to 8.4% (+5.9). This shows that probabilistic embeddings are a simple yet efficient approach for improving unsupervised temporal action segmentation.

2 Related Work

Fully supervised action segmentation approaches yield promising results, but annotating the labels per frame is tedious and time-consuming. While research on weakly-supervised action segmentation requires less labor-intensive effort, unsupervised approaches do not require video labels at all and can scale to large datasets, making them more desirable.

Unsupervised Action Segmentation. Learning to solve a pretext task is a common paradigm within the unsupervised learning literature. Following this trend, Kukleva *et al*. [16] train a model to predict relative timestamps for representation learning, followed by a K-means clustering and HMM to produce final segmentation. In a similar vein, VidalMata *et al*. [34] extend this method to incorporate visual information while the clustering procedure remains the same. Li and Todorovic [18] further improve the representation learning and use an HMM with a length model to tackle this problem and achieve decent results. As pointed out by Kumar *et al*. [17], this paradigm separates representation learning from clustering and as a result, yields sub-optimal segmentation results. They therefore propose a joint representation learning and online clustering method, where a temporal optimal transport is used to estimate pseudo-labels during training. However, the performance of such approaches is limited by the strong fixed action ordering assumption. To address this limitation, Xu and Gould [38] propose a new method for action segmentation called ASOT, which uses a combination of Kantorovic and Gromov-Wasserstein optimal transport for joint representation learning and clustering, the key innovation is that fixed action ordering assumption is no longer enforced, also ASOT does not assume that different actions are evenly distributed across videos by introducing an unbalanced regularization term. This approach yields promising results. More recently, Elena and Dimiccoli [3] extend ASOT by introducing the feedback between frame and segment representations. However, the method contains optimal transport on three levels, which makes the model overly complex. In contrast, we adopt a single optimal transport while focusing on learning a better frame representation by modeling uncertainty within representation learning.

Probabilistic Embeddings. Probabilistic embeddings have been studied in previous works. Vilnis and McCallum [35] use Gaussian embeddings for word representation learning, and the KL divergence between two Gaussians is used as a similarity measure between embeddings. Oh [21] describes a similar idea for learning image embeddings, where the model is trained using a variational information bottleneck objective, in order to handle occlusion in images. Shi and Jain [26] propose probabilistic face embeddings that improve face recognition performance. Sun *et al*. [29] follow [21] and advocate the use of probabilistic embeddings to address 3D-2D human pose ambiguity. They use the learned embeddings for human pose retrieval and action alignment. However, all these methods work in the supervised learning setup, which requires manual labels to

specify whether a pair of observations is matched or not during training, while we address temporal action segmentation and aim to learn probabilistic embeddings in a completely unsupervised manner.

Graph Convolutional Network for Video Understanding. Graph Convolutional Networks (GCN) [13] have been widely used for video understanding. Wang [37] describes a spatial-temporal GCN to model object interactions for action recognition, where 2D bounding boxes across frames serve as nodes and edges model their spatial-temporal interaction. Zeng [41] adopts GCN for action detection. In the context of temporal action segmentation, Huang *et al.* [8] propose a classification and a regression GCN, where initial action segments are treated as nodes of the graph, which successfully improves the baseline segmentation performance [7], particularly in ego-centric cases. More recently, Khan *et al.* [10] use a GCN to address timestamp-supervised action segmentation. Our work on the other hand, tackles a more challenging unsupervised temporal action segmentation problem by using a GCN to learn smooth frame representations and uncertainty, which has not been done before.

3 Approach

Problem Formulation. Given a dataset $\mathcal{D} := \{\mathcal{V}_b\}_{b=1}^{B}$ consisting of B videos, frame-wise embeddings $\mathbf{X} \in \mathbb{R}^{N \times D}$ are extracted by an MLP for each video $\mathcal{V}_b$, where N is the number of video frames and D the dimension of the embedding. We aim to learn new embeddings $\mathbf{Z} \in \mathbb{R}^{N \times D}$, as well as a set of K action prototypes, represented as $\mathbf{A} := [\mathbf{a}_1, \ldots, \mathbf{a}_K] \in \mathbb{R}^{K \times D}$, where $\mathbf{a}_i \in \mathbb{R}^D$ corresponds to the i-th action prototype. In this section, we briefly revisit the optimal transport (OT) formulation [38] for unsupervised action segmentation, then detail our proposed learning pipeline in Sec. 3.2.

3.1 Optimal Transport Formulation

Our work uses the same OT proposed by Xu and Gould [38], which is a combination of Kantorovic Optimal Transport (KOT) [31] and Gromov-Wasserstein Optimal Transport (GWOT) [22].

Kantorovic Optimal Transport. The classical Kantorovich optimal transport is essentially a linear program. Given the cost matrix $\mathbf{C}^k \in \mathbb{R}_+^{N \times K}$ and $\mathbf{p} = \frac{1}{N}\mathbf{1}_N$ and $\mathbf{q} = \frac{1}{K}\mathbf{1}_K$, where $\mathbf{1}_N$ and $\mathbf{1}_K$ are N and K dimensional vectors of ones, the optimization aims to find the minimum assignment $\mathbf{T}^*$:

$$\min_{\mathbf{T} \in \mathcal{T}} \mathcal{F}_{\mathrm{KOT}}(\mathbf{C}^k, \mathbf{T}) := \langle \mathbf{C}^k, \mathbf{T} \rangle, \tag{1}$$

$$\left\{ \mathbf{T} \in \mathbb{R}_+^{N \times K} : \mathbf{T}\mathbf{1}_K = \mathbf{p}, \mathbf{T}^\top \mathbf{1}_N = \mathbf{q} \right\}, \tag{2}$$

where $\mathcal{T}$ represents a set of possible transportation polytopes. In the context of temporal action segmentation, $\mathbf{C}^k$ can be interpreted as the cost of assigning N frames to K actions.

Gromov-Wasserstein Optimal Transport. The Gromov-Wasserstein optimal transport extends the Kantorovich formulation by allowing for the comparison of histograms defined over incomparable spaces. Given two metric-measure pairs $(\mathbf{C}^v, \mathbf{p})$ and $(\mathbf{C}^a, \mathbf{q})$, the objective function is defined as:

$$\mathcal{F}_{\mathrm{GW}}(\mathbf{C}^v, \mathbf{C}^a, \mathbf{T}) := \sum_{i,k\in[n],j,l\in[m]} L(c^v_{ik}, c^a_{jl}) t_{ij} t_{kl}, \tag{3}$$

where $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ measures the discrepancies between the cost matrices, and $[n]$ and $[m]$ denote the set of video frames and action embeddings.

In order to model the long-tail nature of action segments, ASOT [38] further relaxes the second hard constraint in Eq. (1) into a soft constraint by minimizing the KL divergence between action marginals and a uniform action distribution across frames, where a small λ encourages a more unbalanced solution $\mathbf{T}$ [4, 5]. Finally, an entropy regularization term is added to the objective. Therefore, the final OT objective is defined as:

$$\begin{aligned} \min_{\mathbf{T}\in\mathcal{T}} \quad & \alpha\mathcal{F}_{\mathrm{GW}}(\mathbf{C}^v, \mathbf{C}^a, \mathbf{T}) + (1-\alpha)\mathcal{F}_{\mathrm{KOT}}(\mathbf{C}^k, \mathbf{T}) + \\ & \lambda D_{\mathrm{KL}}(\mathbf{T}^\top \mathbf{1}_N \parallel \mathbf{q}) - \epsilon H(\mathbf{T}), \end{aligned} \tag{4}$$

with the solution satisfying $\{\mathbf{T} \in \mathbb{R}_+^{N\times K} : \mathbf{T}\mathbf{1}_K = \mathbf{p}\}$. This non-convex optimization problem is solved using the efficient mirror descent [22, 38] algorithm with time complexity $\mathcal{O}(NK)$ per iteration.

Cost Matrices. Eq. (4) contains a set of cost matrices, $\mathbf{C} := \{\mathbf{C}^k, \mathbf{C}^v, \mathbf{C}^a\}$ for the KOT and GWOT problem. Specifically, $\mathbf{C}^k$ is the visual component and is defined as $c^k_{ij} = 1 - \frac{\mathbf{x}_i^\top \cdot \mathbf{a}_j}{\|\mathbf{x}_i\| \|\mathbf{a}_j\|} + \rho \cdot r_{ij}$ and $\mathbf{R} \in \mathbb{R}^{N\times K}$ is the temporal prior commonly used in [17, 38] defined as $r_{ij} = |i/N - j/K|$ for $\rho \geq 0$.

The cost matrices $\mathbf{C}^v \in \mathbb{R}^{N\times N}$ and $\mathbf{C}^a \in \mathbb{R}^{K\times K}$ serve as structural prior by penalizing associating adjacent frames ($|i-k| \leq Nr, i \neq k$) to different actions ($j \neq l$) for two different assignments t_{ij} and t_{kl}. However, no penalty is applied to assignments outside the temporal radius Nr or if adjacent frames are mapped to the same action ($j = l$).

3.2 Probabilistic Embeddings

Prior unsupervised segmentation works [3, 17, 32, 38] essentially learn deterministic embeddings as frame representations, which are used to obtain pseudo-labels via OT. However, networks trained with deterministic embeddings can quickly overfit to the data. We argue that it is more natural to learn a probabilistic embedding rather than its deterministic counterpart by incorporating the data-dependent (heteroscedastic) uncertainty [9] into the embeddings. To this end, we propose to parametrize the embeddings such that they follow a Gaussian distribution with a diagonal covariance matrix:

$$p(\mathbf{z}|\mathbf{x}_i) = \mathcal{N}(\boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2)). \tag{5}$$

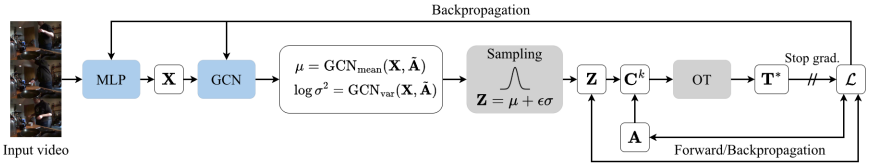


Fig. 2: Pipeline of our training scheme. Features of an input video are fed through an MLP to obtain per-frame embeddings $\mathbf{X}$. A temporally weighted graph is constructed, and the normalized adjacency matrix $\tilde{\mathbf{A}}$ along with frame embeddings $\mathbf{X}$ are fed into a Graph Convolutional Network (GCN), which produces the mean and covariance of the frame embeddings. The re-parametrization trick is used during training to obtain probabilistic frame embeddings $\mathbf{Z}$, along with trainable action embeddings $\mathbf{A}$, to produce the cost $\mathbf{C}^k$ for OT that finds the pseudo-label $\mathbf{T}^*$. We optimize the cross-entropy loss between the framewise probability distribution and the pseudo-label $\mathbf{T}^*$. **Blue** boxes indicate the **architectural components** of the model that are trainable, and arrows denote computation/gradient flow.

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ are both D-dimensional vectors predicted by the network for the i-th frame $\mathbf{x}_i$. In particular, we sample a standard Gaussian noise ϵ during training and use the differentiable re-parametrization trick [12] to sample the frame embeddings in the forward pass, *i.e.*, $\mathbf{z}_i = \boldsymbol{\mu}_i + \epsilon\boldsymbol{\sigma}_i$. The sampled embeddings $\mathbf{z}$ are then used to estimate pseudo-labels from OT. Due to the sampling, more variations of the pseudo-labels are generated during training. While the loss is at the beginning higher compared to a deterministic embedding, it converges in general to a better solution as shown in the supplementary material. When the embedding is trained, we use only $\boldsymbol{\mu}_i$ for inference, *i.e.*, the probabilistic embedding does not increase the inference time compared to an approach with a deterministic embedding.

Predicting Probabilistic Embeddings In order to integrate temporal context into probabilistic modeling, we propose to use Graph Convolutional Networks (GCN) [13] to learn smooth representations and uncertainty as shown in Fig. 2. Given an input video $\mathcal{V}$, we first feed it into an MLP ϕ to get per-frame embeddings $\mathbf{X} \in \mathbb{R}^{N \times D}$, where D is the dimension of the embedding. Inspired by [24, 37], we construct a temporally weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each frame represents a node and the edges connect every two adjacent frames. The edge weight a_{ij} in the adjacency matrix $\mathbf{A}_{\text{adj}} \in \mathbb{R}^{N \times N}$ is defined as the cosine similarity of pairwise features

$$a_{ij} = \frac{\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)}{||\phi(\mathbf{x}_i)||^2 ||\phi(\mathbf{x}_j)||^2}. \tag{6}$$

We show in the ablation study that using a weighted adjacency matrix works better than using an unweighted adjacency matrix. We then add self-connections to the adjacency matrix, *i.e.*, $\hat{\mathbf{A}} = \mathbf{A}_{\text{adj}} + \mathbf{I}$ where $\mathbf{I} \in \mathbb{R}^{N \times N}$ is the identity matrix. This self-connected adjacency matrix is further normalized via $\tilde{\mathbf{A}} = \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}}$ where $\hat{\mathbf{D}}$ is the degree matrix of $\hat{\mathbf{A}}$. In particular, given the input

representation $\mathbf{X}$ and the normalized adjacency matrix $\tilde{\mathbf{A}}$, the output of one GCN layer is computed as:

$$\mathbf{Z} = \sigma(\tilde{\mathbf{A}}\mathbf{X}\mathbf{W}) \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{D\times D}$ denotes the weights of the GCN network while σ is the activation function and $\mathbf{Z} \in \mathbb{R}^{N\times D}$. GCN makes it possible for the model to incorporate inductive bias in videos. As a result, we use GCN to predict the probabilistic embeddings:

$$\boldsymbol{\mu} = \mathrm{GCN}_{\mathrm{mean}}(\mathbf{X}, \tilde{\mathbf{A}}), \log \boldsymbol{\sigma}^2 = \mathrm{GCN}_{\mathrm{var}}(\mathbf{X}, \tilde{\mathbf{A}}), \tag{8}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{N\times D}$ and $\boldsymbol{\sigma} \in \mathbb{R}^{N\times D}$ indicate the mean and covariance of the predicted Gaussian distribution for each frame. The network estimates the log of the variance to ensure positivity. We then use $\mathbf{Z} = \boldsymbol{\mu} + \epsilon\boldsymbol{\sigma}$ during training where ϵ is the noise sampled from standard Gausssian, to estimate pseudo-labels via OT. We will show in the ablation studies that computing $\mathbf{Z}$ using a GCN leads to better embeddings than using an MLP or TCN.

Training. Previous works [3,17,32,38] adopt a joint representation learning and clustering paradigm, where the methods alternate between using OT to generate pseudo-labels and optimizing the standard cross-entropy loss using gradient descent, following the Expectation-Maximization (EM) algorithm [6]. Given a frame i and its deterministic frame embedding $\mathbf{x}_i$, the probability of assigning frame i to action j is defined as $p_{ij} = \frac{\exp(\mathbf{x}_i^T \mathbf{a}_j/\tau)}{\sum_l \exp(\mathbf{x}_i^T \mathbf{a}_l/\tau)}$ where τ is the temperature. Together with the pseudo-label $\mathbf{T} \in \mathbb{R}^{N\times K}$ derived from OT, the cross-entropy loss is defined as:

$$\mathcal{L}_{\mathrm{ce}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} t_{ij} \log p_{ij}. \tag{9}$$

However, such modeling assumes a single "winner-takes-all" pseudo-label; models can be over-confident with this pseudo-label and may overfit to the noise inherent in the data. In order to address this problem, we further propose to optimize a novel uncertainty-based cross-entropy loss:

$$\mathcal{L}_{\mathrm{uncer}} = \mathbb{E}_{p_\phi(\mathbf{Z}|\mathbf{X})}\mathcal{L}_{\mathrm{ce}}. \tag{10}$$

By using probabilistic embeddings $\mathbf{Z}$ rather than its deterministic counterparts $\mathbf{X}$, the algorithm essentially considers different pseudo-labels during training, to avoid the model from overfitting to noisy pseudo-labels $\mathbf{T}$. As it is intractable to integrate the expectation in Eq. (10), we propose to approximate the uncertainty loss via Monte Carlo sampling. Concretely, we sample M times from the Gaussian distribution during training to obtain the sampled features and the corresponding cross-entropy losses, followed by averaging them to approximate Eq. (10):

$$\mathcal{L}_{\text{uncer}} \approx -\frac{1}{MN} \sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{j=1}^{K} t_{ij}^{(m)} \log p_{ij}^{(m)}. \tag{11}$$

During inference, we simply take $\boldsymbol{\mu} = \text{GCN}_{\text{mean}}(\mathbf{X}, \tilde{\mathbf{A}})$ as the final frame embeddings $\mathbf{Z} \in \mathbb{R}^{N \times D}$ to calculate the cost in $\mathbf{C}^k$ and obtain the final segmentation using OT as described in Section 3.1.

Computational Complexity. In comparison to prior works [2,38] which relies on a single pseudo label per EM iteration, our approach requires M pseudo labels during each training iteration and our training complexity is thus $\mathcal{O}(MTNK)$ compared to $\mathcal{O}(TNK)$ where T is the number of steps for gradient descent. Since M is small during training, it adds only a very small additional computational cost as shown in the supplementary material.

4 Experiments

4.1 Experimental Setup

Implementation Details. Following [38], we design the encoder MLP with a single hidden layer and ReLU activation. We use a one-layer GCN with a graph connecting every 3 neighboring frames. For training, we sample $M = 3$ times from the Gaussian distributions. Adam optimizer [11] is adopted with a learning rate of 10^{-3} for the representation learning with a weight decay of 10^{-4}. We use K-means clustering to initialize the action embeddings, where K equals the ground truth number of actions per activity [3, 16–18, 32, 38]. For each video, we sample 256 frames from uniformly distributed intervals [17, 38]. We provide more hyperparameter setting details in the supplementary material.

Datasets and Features. We conduct experiments on four datasets. For each dataset, we use the same pre-extracted features for training and inference, consistent with prior works [3, 16–18, 38].

- **Breakfast (BF)** [14] is a large-scale dataset that includes 10 cooking activities. Each video spans a few seconds to several minutes, with multiple actions, where actions can be performed in a different temporal order, for example, `add teabag` can happen before/after `pour water` action across videos in the `Tea` activity. Also, repetitive actions such as `squeeze orange` and `pour juice` in the `Juice` activity can occur multiple times within a video, making it particularly challenging for unsupervised action segmentation. This dataset has a total of 48 actions across 1712 videos. We use the IDT [36] features.
- **Youtube Instructional (YTI)** [1] has 5 activities and each activity has 30 videos. The average per-video length is roughly 2 minutes, with a large portion of frames being background frames. We use the features provided by [1].

- **50Salads (FS)** [28] comprises 50 videos of actors making salad. Following prior works [3, 16, 18, 38], we evaluate our model on the Eval granularity, which contains 12 action classes, where actions such as `cut tomato` and `cut cheese` are treated as a single `cut` action. IDT [36] features are used as input.
- **Desktop Assembly (DA)** [17] has 76 videos of actors performing one assembly activity, where each video is roughly 1.5 minutes long. Each actor conducts 22 actions in a fixed temporal order. We use the features shared by [17].

Evaluation Metrics. We use the evaluation protocol proposed by [16, 25] for evaluating unsupervised action segmentation. We match the predicted segmentations with the ground truth segmentations via the Hungarian algorithm [15] across all videos of the same activity, as done by [3,16–18,25,32,38]. We use three metrics, namely mean-over-frames (MoF), F1-score, and mean intersection-over-union (mIoU). MoF calculates the percentage of correct per-frame predictions. F1-score is defined on the per-segment level, *i.e.*, for a predicted segment that is matched to a ground truth segment. If the number of correct frames exceeds 50% of the ground truth segment length, it is regarded as a true positive segment [16]. mIoU averages IoU over all classes. We report F1 and mIoU averaged across all activities for a dataset.

4.2 Comparison with State of the Art

We name our unsupervised segmentation approach as probabilistic embeddings optimal transport (PEOT) and evaluate it on Breakfast [14], Youtube Instr. [1], 50Salads (Eval) [28], as well as Desktop Assembly [17]. The results are summarized in Table 1. Our method outperforms the state of the art for 9 out of 12 metrics and datasets. On Breakfast and Desktop Assembly, our approach outperforms the state of the art for all metrics. On Youtube Instr. and 50Salads (Eval), the very recent works VASOT [2] and CLOT [3], which are complementary extensions of ASOT, report better results for some metrics, whereas our approach achieves the highest MoF. VASOT [2] is a recently proposed method that jointly solves action alignment and action segmentation by extending the fused Gromov-Wasserstein optimal transport to match a pair of videos from the same activity. It surpasses ASOT at the cost of increased computational complexity. Our approach consistently yields better results than VASOT in terms of MoF and F1, despite having a much simpler methodology design. CLOT [3] is another recent extension to ASOT. It introduces feedback between frame-wise and segment-wise representations via the cross-attention [33] mechanism. The system also integrates a projection-based sliced Wasserstein distance [20] that successfully detects small segments, a significant limitation of ASOT. On the Breakfast and Desktop Assembly dataset, our approach outperforms CLOT on all metrics and it performs comparable than CLOT on the Youtube Instr. and 50Salads (Eval) datasets with lower mIoU and higher MoF.

The gain compared to ASOT [38] is most important since it is our baseline. MoF compared to ASOT is increased by +4.6 (8.2%) on Breakfast, +1.5 (4.7%) on YTI, +5.6 (9.4%) on 50Salads, and +0.8 (1.1%) on Desktop Assembly, and

Table 1: Comparisons of action segmentation performance obtained by applying the Hungarian matching at the activity-level on the Breakfast [14], Youtube Instr. [1], 50Salads (Eval) [28] and Desktop Assembly [17] datasets. The highest accuracy is indicated in **bold**, and the second highest is underlined.

Methods	Breakfast			YTI			50Salads (Eval)			DA		
	MoF	F1	mIoU	MoF	F1	mIoU	MoF	F1	mIoU	MoF	F1	mIoU
CTE [16]	41.8	26.4	-	39.0	28.3	-	35.5	-	-	47.6	44.9	-
VTE [34]	48.1	-	-	-	29.9	-	30.6	-	-	-	-	-
UDE [30]	47.4	31.9	-	43.8	29.6	-	42.2	34.4	-	-	-	-
ASAL [18]	52.5	37.9	-	44.9	32.1	-	39.2	-	-	-	-	-
TOT [17]	47.5	31.0	-	40.6	30.0	-	47.4	42.8	-	56.3	51.7	-
TOT+ [17]	39.0	30.3	-	45.3	32.9	-	44.5	48.2	-	58.1	53.4	-
UFSA [32]	52.1	38.0	-	49.6	32.4	-	55.8	50.3	-	65.4	63.0	-
ASOT [38]	56.1	38.3	18.6	52.9	32.1	24.7	59.3	53.6	30.1	70.4	68.0	45.9
HVQ [27]	54.4	39.7	-	50.3	35.1	-	-	-	-	-	-	-
VASOT [2]	57.5	39.0	18.8	53.2	35.7	**25.2**	60.6	57.4	34.5	70.9	75.1	49.3
CLOT [3]	60.1	40.1	18.5	54.4	36.7	23.4	59.4	**63.2**	**38.8**	68.8	72.6	48.1
PEOT (Ours)	**60.7**	**40.5**	**19.0**	**55.4**	**37.4**	22.9	**64.9**	58.9	30.2	**71.2**	**75.8**	**51.7**

Table 2: Effect of probabilistic embeddings on the four datasets for ASOT and VASOT: Breakfast [14], Youtube Instr. [1], 50Salads (Eval) [28], and Desktop Assembly [17]. The results of the baselines have been computed using the public available source codes.

	Breakfast			YTI			50Salads (Eval)			DA		
Methods	MoF	F1	mIoU	MoF	F1	mIoU	MoF	F1	mIoU	MoF	F1	mIoU
ASOT [38]	56.4	35.7	17.2	49.0	34.1	23.7	59.4	56.7	25.6	59.0	63.7	40.6
ASOT [38] + Prob.	60.7	40.5	19.0	55.4	37.4	22.9	64.9	58.9	30.2	71.2	75.8	51.7
VASOT [2]	54.5	35.3	15.5	47.5	30.4	18.0	53.4	51.9	26.7	67.9	70.2	48.0
VASOT [2] + Prob.	57.2	36.1	15.8	52.5	32.5	18.8	62.2	53.6	26.4	71.1	76.1	51.8

the F1-score is increased by +2.2 (5.7%) on Breakfast, +5.3 (16.5%) on YTI, +5.3 (9.9%) on 50Salads, and +7.8 (11.5%) on Desktop Assembly.

While we compare in Table 1 our approach to results that have been reported in the literature, we also provide a direct comparison to ASOT [38] and VASOT [2] using the public available source code in Table 2. Since the source code of CLOT [3] is only partially available, we could not include it in the comparison. To demonstrate that probabilistic embeddings improve deterministic embeddings, we added them to ASOT [38] and VASOT [2] as baselines. In both cases, we observe improvements with the largest gains on the Desktop Assembly dataset. Compared to ASOT, our approach improves MoF up to 20.7% (+12.2) and F1-score up to 19.0% (+12.1). Compared to VASOT, our approach improves MoF up to 16.5% (+8.8) and F1-score up to 8.4% (+5.9). This shows that probabilistic embeddings are a simple yet efficient approach for improving unsupervised temporal action segmentation. We provide qualitative results in Section 4.4.

Table 3: Impact of probabilistic embedding and GCN on the four datasets: Breakfast [14], Youtube Instr. [1], 50Salads (Eval) [28] and Desktop Assembly [17].

		Breakfast			YTI			50Salads (Eval)			DA		
Prob.	GCN	MoF	F1	mIoU	MoF	F1	mIoU	MoF	F1	mIoU	MoF	F1	mIoU
		56.4	35.7	17.2	49.0	34.1	23.7	59.4	56.7	25.6	59.0	63.7	40.6
✓		59.2	34.9	15.1	49.4	33.7	20.0	60.1	56.3	23.7	63.8	63.9	42.3
	✓	58.2	40.7	18.5	48.9	33.9	22.9	50.4	48.7	17.2	63.8	70.5	46.2
✓	✓	60.7	40.5	19.0	55.4	37.4	22.9	64.9	58.9	30.2	71.2	75.8	51.7

Table 4: Comparison of MLP, TCN, and GCN on Breakfast [14].

Methods	MoF	F1	mIoU
MLP	59.2	34.9	15.1
TCN	59.2	38.9	17.9
GCN	60.7	40.5	19.0

Table 5: Ablation study of graph connectivity for GCN on Breakfast [14].

Connectivity	MoF	F1	mIoU
5 frames	58.9	40.0	18.5
3 frames	60.7	40.5	19.0

Table 6: Impact of weighted adjacency matrix for GCN on Breakfast [14].

Methods	MoF	F1	mIoU
Unweighted adjacency matrix	59.5	40.2	18.0
Weighted (learned) adjacency matrix	60.7	40.5	19.0

Table 7: Impact of number of samples on Breakfast [14].

Samples	MoF	F1	mIoU
$M = 1$	60.0	39.4	18.4
$M = 2$	60.3	40.1	18.8
$M = 3$	60.7	40.5	19.0
$M = 5$	60.6	40.3	18.9

4.3 Ablation Study

Impact of Probabilistic Embeddings and GCN. We study the impact of different components on our system in Table 3. If we do not use probabilistic embeddings and a GCN to estimate them, our approach is the same as ASOT [38] since we use the same MLP architecture and OT formulation as ASOT. If we learn the probabilistic embeddings using an MLP (row 2 of Table 3), *i.e.*, by parameterizing $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}^2$ of the Gaussian distributions with one added MLP layer, it consistently improves MoF on all datasets. However, it does not improve F1 and mIoU, it even decreases these metrics in most cases. Since the MLP estimates the Gaussian distributions using the features of only one frame, the estimates are not very reliable. If we learn the probabilistic embeddings using a GCN (row 4) instead of an MLP, we get except of mIoU on YTI a substantial improvement for all metrics and datasets compared to the baseline (row 1). The MoF is improved by +4.3 (7.6%), +6.4 (13.1%), +5.5 (9.3%), and +12.2 (20.7%) for the Breakfast, YTI, 50Salads, and Desktop Assembly datasets, respectively. The F1-score is improved by +4.8 (13.4%), +3.3 (9.7%), +2.2 (3.9%), and +12.1 (19.0%). In particular the improvements over the baseline for MoF and F1-score

Table 8: Comparison with other stochastic regularization approaches: Gauss: GCN+fix Gaussian noise, Drop: GCN+Dropout, Ours: GCN+Learned uncertainty.

	Breakfast			YTI			50Salads (Eval)			DA		
Stoch.	MoF	F1	mIoU	MoF	F1	mIoU	MoF	F1	mIoU	MoF	F1	mIoU
Gauss	58.3	38.7	18.1	53.5	34.9	21.3	61.0	58.8	30.2	61.9	63.2	42.0
Drop	58.5	40.7	18.4	48.5	34.3	22.3	62.0	58.6	29.6	65.3	73.8	48.6
Ours	60.7	40.5	19.0	55.4	37.4	22.9	64.9	58.9	30.2	71.2	75.8	51.7

are very large. We show that these improvements are not only due to the GCN (row 3). While the GCN improves the baseline as well in many cases, the largest and most consistent improvements are due to the probabilistic embeddings.

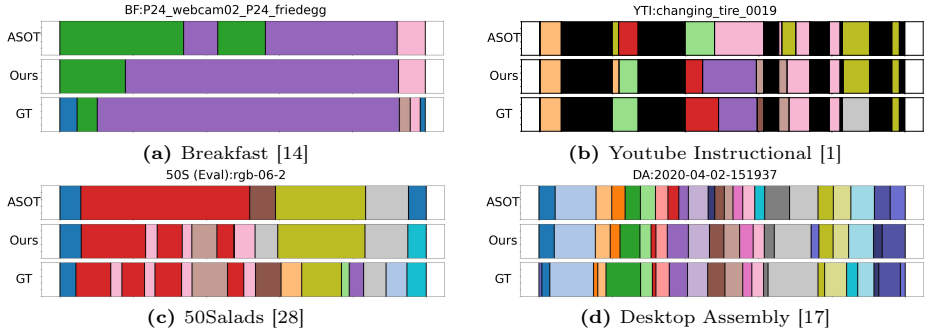
In Table 4, we also evaluate the impact of replacing the GCN layer by a TCN [7] layer, *i.e.*, a 1-dimensional temporal convolution of kernel size 3. While TCN performs better than an MLP, the GCN layer performs best. While TCN uses a convolution kernel with fixed weights over all temporal frames, the GCN layer adaptively adjusts the weights based on the input features.

Impact of Graph Connectivity in GCN. We investigate the impact of adding more connections to build the graph $\mathcal{V}$ for the GCN. By default, each node is connected to three nodes, *i.e.*, the node at frame i is connected to the nodes at frames $i-1$, i, and $i+1$. We tried more connections such that the node at frame i is connected to the nodes at frames $i-2$ and $i+2$ as well, but the results in Table 5 show that increasing the number of connections does not increase the performance. This might be attributed to the fact that adding more edges in the graph leads to the over-smoothing problem in GCNs [13, 41]. For this reason, we use the 3-frames connection graph for the GCN.

Impact of Weighted Adjacency Matrix in GCN. We investigate the benefit of using a weighted adjacency matrix (6) in our GCN in Table 6. For comparison, we use a GCN layer without weighted adjacency matrix, *i.e.*, a_{ij} in $\mathbf{A}_{\text{adj}}$ is 1 if there is a connection between frame i and j and 0 otherwise. We notice that using a weighted adjacency matrix for the GCN leads to better performance.

Impact of M. We study the effect of using a different number of Monte Carlo samples during training (11) in Table 7. Note that for $M = 1$ we still have a probabilistic embedding, but we sample only once from the learned Gaussian distributions. We notice that increasing M from 1 to 3 increases the performance, but it saturates after $M = 3$.

Comparisons of Probabilistic Embeddings with Stochastic Regularizations. We compare to stochastic regularization approaches in Table 8, *i.e.*, adding Gaussian noise to deterministic embeddings or adding dropout to the GCN. All methods use the same GCN architecture for fair comparison. Except of F1 on Breakfast, where dropout performs slightly better, our approach outperforms the stochastic regularization methods. For instance, F1 is +2.5 (7.2%) higher on YTI and MoF is +5.9 higher (9%) on DA. It demonstrates that our approach is more effective than other stochastic regularization approaches.



(a) Breakfast [14] **(b)** Youtube Instructional [1]

(c) 50Salads [28] **(d)** Desktop Assembly [17]

Fig. 3: Qualitative results. Comparing ASOT [38], PEOT (ours), and ground truth (GT) across different datasets and activities.

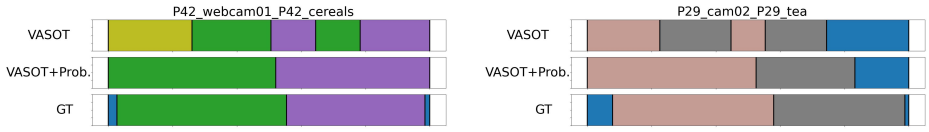
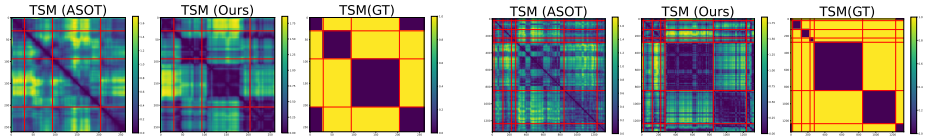


Fig. 4: Qualitative results. Two examples when using VASOT [2] as baseline. The examples are from the Breakfast [14] dataset.

4.4 Qualitative Results

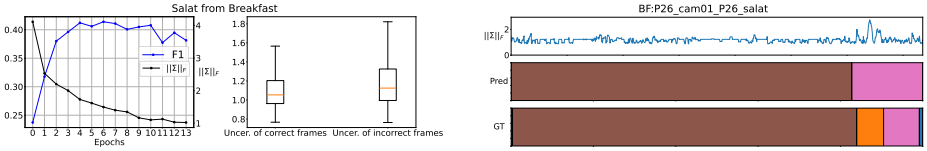
We finally provide some qualitative segmentation results in Fig. 3, comparing our approach to the ASOT baseline. In the example from the Breakfast dataset, our method is able to identify the long purple segment as opposed to the baseline approach which splits it into several segments. In the example from YTI, our method successfully segments very short actions. While ASOT finds good segments as well, more segments are not correctly clustered, *i.e.*, not associated to the correct action. The black color indicates background frames. In the example from 50Salads, our method can detect reoccurring actions whereas ASOT does not recognize that some actions reoccur in this case. In the example from the Desktop Assembly dataset, the segments estimated by our approach are better aligned with the ground-truth. We further show some qualitative results for adding probabilistic embeddings to VASOT [2] as baseline in Fig. 4. In these examples, VASOT generates additional segments for the green and gray cluster. With the probabilistic embeddings, the segments are better aligned with the ground truth.

Qualitative Results of Learned Frame Embeddings. We visualize the learned frame embeddings by calculating the temporal self-similarity matrix for a given video, where 1 minus the cosine similarity between every two frames is computed. We see in Fig. 5 that our approach gives better embeddings than ASOT [38], which uses a deterministic embedding. In the first example our learned representations are more temporally coherent even for short segments, although it is slightly misaligned with the ground truth action boundaries. For



(a) Breakfast: P12_webcam01_P12_tea **(b)** Breakfast: P15_stereo01_P15_juice

Fig. 5: Qualitative results of the learned frame-wise representations. Comparing ASOT [38] (left), our approach (middle), and ground truth (right) in terms of the temporal self-similarity matrix. Our approach learns better representations that are closer aligned to the ground truth segmentation.



(a) Uncertainty progress during training (left) and statistics of frame-wise uncertainty (right). **(b)** Qualitative result of segmentation and uncertainty estimation for one video of Breakfast.

Fig. 6: Analysis of the learned uncertainty.

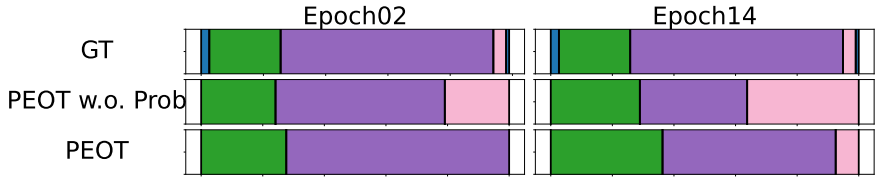


Fig. 7: Difference between probabilistic embeddings and deterministic embeddings during training. After 2 epochs, the approach using the deterministic embedding (row 2) recognizes the pink action, while the probabilistic embedding (row 3) misses it. After 14 epochs, the segmentation with the deterministic embedding only changes by shifting the boundaries between the segments whereas the pink action now also appears for the probabilistic embedding. The segmentation of the latter is now also closer to the ground truth. The video is `P52_stereo01_P52_friedegg` taken from Breakfast.

the large blue squares in the ground truth matrix of the second example, ASOT shows stronger dissimilar patterns in the self-similarity matrix, which is an indicator that the learned representation focuses too much on subtle details that are not relevant for distinguishing actions.

4.5 Analysis of the Learned Uncertainty.

We report the progress of the learned covariance during training in Fig. 6, using the Frobenius norm $||\Sigma||_F$ to measure the frame-wise uncertainty. In Fig. 6a, we average the uncertainty across all frames of the `Salat` activity of Breakfast. The uncertainty is high at the beginning of the training. As the training progresses,

the learned features become better and the uncertainty decreases. It might be beneficial to sample more often at the beginning of the training when the uncertainty is high, which we leave as a future work. We also observe that a low uncertainty is an indicator for overfitting. We report a box plot (right of Fig. 6a) for the uncertainties of all correctly predicted vs. all incorrectly predicted frames after training. It shows that the estimated uncertainty is in general higher for incorrectly predicted frames, but some incorrectly predicted frames have a low uncertainty. This is visualized in Fig. 6b. For the missed orange segment, the peaks of the uncertainty are higher compared to the correctly detected segments, but the uncertainty is not high for all frames within the orange segment.

We further plot the evolution of the results for two different training epochs in Fig. 7. The example shows that the deterministic embedding provides a better segmentation at the early epoch, but the probabilistic embedding changes more during training. At the later epoch, the segmentation of the probabilistic embedding is closer to the ground-truth.

5 Conclusion

In this work, we proposed learning probabilistic instead of deterministic embeddings for frame representations, which brings a novel perspective to unsupervised temporal action segmentation. We incorporated the approach into two state-of-the-art baselines, namely ASOT [38] and VASOT [2], and evaluated it on four challenging benchmarks. The results showed that the probabilistic embeddings consistently improved MoF and F1-score over all datasets. Compared to the state of the art, our approach achieves the best performance for 9 out of 12 metrics/datasets. Future work can make the pseudo label generation process differentiable [19] and the probabilistic embeddings might be also useful for other tasks like action anticipation [40].

Acknowledgements

The work has been supported by the project iBehave (receiving funding from the programme “Netzwerke 2021”, an initiative of the Ministry of Culture and Science of the State of Northrhine Westphalia) and the ERC Consolidator Grant FORHUE (101044724). The authors would like to thank Elena Bueno-Benito, Federico Spurio and Yazan Abu Farha for helpful discussions.

References

1. Alayrac, J.B., Bojanowski, P., Agrawal, N., Sivic, J., Laptev, I., Lacoste-Julien, S.: Unsupervised learning from narrated instruction videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4575–4583 (2016)
2. Ali, A.S., Mahmood, S.A., Saeed, M., Konin, A., Zia, M.Z., Tran, Q.H.: Joint self-supervised video alignment and action segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10807–10818 (2025)

3. Bueno-Benito, E., Dimiccoli, M.: Clot: Closed loop optimal transport for unsupervised action segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10719–10729 (2025)
4. Chizat, L., Peyré, G., Schmitzer, B., Vialard, F.X.: Scaling algorithms for unbalanced optimal transport problems. Mathematics of computation **87**(314), 2563–2609 (2018)
5. De Plaen, H., De Plaen, P.F., Suykens, J.A., Proesmans, M., Tuytelaars, T., Van Gool, L.: Unbalanced optimal transport: A unified framework for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3198–3207 (2023)
6. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society: series B (methodological) **39**(1), 1–22 (1977)
7. Farha, Y.A., Gall, J.: Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3575–3584 (2019)
8. Huang, Y., Sugano, Y., Sato, Y.: Improving action segmentation via graph-based temporal reasoning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14024–14034 (2020)
9. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? Advances in neural information processing systems **30** (2017)
10. Khan, H., Haresh, S., Ahmed, A., Siddiqui, S., Konin, A., Zia, M.Z., Tran, Q.H.: Timestamp-supervised action segmentation with graph convolutional networks. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10619–10626. IEEE (2022)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
13. Kipf, T.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
14. Kuehne, H., Arslan, A., Serre, T.: 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)
15. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
16. Kukleva, A., Kuehne, H., Sener, F., Gall, J.: 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)
17. Kumar, S., Haresh, S., Ahmed, A., Konin, A., Zia, M.Z., Tran, Q.H.: 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)
18. Li, J., Todorovic, S.: 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)
19. Li, S., Kong, Y., Rezatofighi, H.: Learning of global objective for network flow in multi-object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8855–8865 (2022)
20. Nguyen, K., Ho, N.: Energy-based sliced wasserstein distance. Advances in Neural Information Processing Systems **36**, 18046–18075 (2023)

21. Oh, S.J., Murphy, K., Pan, J., Roth, J., Schroff, F., Gallagher, A.: Modeling uncertainty with hedged instance embedding. arXiv preprint arXiv:1810.00319 (2018)
22. Peyré, G., Cuturi, M., Solomon, J.: Gromov-wasserstein averaging of kernel and distance matrices. In: International conference on machine learning. pp. 2664–2672. PMLR (2016)
23. Romeo, L., Marani, R., Perri, A.G., Gall, J.: Multi-modal temporal action segmentation for manufacturing scenarios. Engineering Applications of Artificial Intelligence **148** (2025)
24. Sarfraz, S., Murray, N., Sharma, V., Diba, A., Van Gool, L., Stiefelhagen, R.: Temporally-weighted hierarchical clustering for unsupervised action segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11225–11234 (2021)
25. Sener, F., Yao, A.: Unsupervised learning and segmentation of complex activities from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8368–8376 (2018)
26. Shi, Y., Jain, A.K.: Probabilistic face embeddings. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6902–6911 (2019)
27. Spurio, F., Bahrami, E., Francesca, G., Gall, J.: Hierarchical vector quantization for unsupervised action segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 6996–7005 (2025)
28. Stein, S., McKenna, S.J.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. pp. 729–738 (2013)
29. Sun, J.J., Zhao, J., Chen, L.C., Schroff, F., Adam, H., Liu, T.: View-invariant probabilistic embedding for human pose. In: European Conference on Computer Vision. pp. 53–70. Springer (2020)
30. Swetha, S., Kuehne, H., Rawat, Y.S., Shah, M.: Unsupervised discriminative embedding for sub-action learning in complex activities. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 2588–2592. IEEE (2021)
31. Thorpe, M.: Introduction to optimal transport. Notes of Course at University of Cambridge **3** (2018)
32. Tran, Q.H., Mehmood, A., Ahmed, M., Naufil, M., Zafar, A., Konin, A., Zia, Z.: 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)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
34. VidalMata, R.G., Scheirer, W.J., Kukleva, A., Cox, D., Kuehne, H.: Joint visual-temporal embedding for unsupervised learning of actions in untrimmed sequences. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1238–1247 (2021)
35. Vilnis, L., McCallum, A.: Word representations via gaussian embedding. arXiv preprint arXiv:1412.6623 (2014)
36. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE international conference on computer vision. pp. 3551–3558 (2013)
37. Wang, X., Gupta, A.: Videos as space-time region graphs. In: Proceedings of the European conference on computer vision (ECCV). pp. 399–417 (2018)

38. Xu, M., Gould, S.: 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)
39. Yang, Y., Li, Y., Fermuller, C., Aloimonos, Y.: Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web. In: Proceedings of the AAAI conference on artificial intelligence. vol. 29 (2015)
40. Zatsarynna, O., Bahrami, E., Farha, Y.A., Francesca, G., Gall, J.: Gated temporal diffusion for stochastic long-term dense anticipation. In: European Conference on Computer Vision. pp. 454–472. Springer (2024)
41. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 7094–7103 (2019)