Leveraging Action Affinity and Continuity for Semi-supervised Temporal Action Segmentation

Guodong Ding[0000−0001−6080−5220] and Angela Yao[0000−0001−7418−6141]

National University of Singapore
{dinggd, ayao}@comp.nus.edu.sg

Abstract. We present a semi-supervised learning approach to the temporal action segmentation task. The goal of the task is to temporally detect and segment actions in long, untrimmed procedural videos, where only a small set of videos are densely labelled, and a large collection of videos are unlabelled. To this end, we propose two novel loss functions for the unlabelled data: an action affinity loss and an action continuity loss. The action affinity loss guides the unlabelled samples learning by imposing the action priors induced from the labelled set. Action continuity loss enforces the temporal continuity of actions, which also provides frame-wise classification supervision. In addition, we propose an Adaptive Boundary Smoothing (ABS) approach to build coarser action boundaries for more robust and reliable learning. The proposed loss functions and ABS were evaluated on three benchmarks. Results show that they significantly improved action segmentation performance with a low amount (5% and 10%) of labelled data and achieved comparable results to full supervision with 50% labelled data. Furthermore, ABS succeeded in boosting performance when integrated into fully-supervised learning.

1 Introduction

Temporal action segmentation aims to segment long, untrimmed procedural video sequences into multiple actions and assign semantic labels for each frame. This task requires the arduous collection of frame-wise labelling for minutes-long videos. Previous works have reduced the annotation effort via weaker supervision in the form of transcripts [15], action sets [10], and timestamp labels [21]. With each method, annotators are still required to watch or scrub through each video in the training set to provide labels for *every* training video. Different from the previous methods, we work under a semi-supervised setting where frame-wise annotations are provided for only a small portion (5% and 10%) of the videos in the training set, while the remaining videos are unlabelled. This setting greatly reduces the annotation efforts.

Semi-supervised learning has been studied extensively in image-based vision tasks such as image classification [23], object detection [33], semantic segmentation [11], *etc.* Two popular semi-supervised learning techniques are consistency regularization [25, 31] and pseudo-labelling [18]. Consistency regularization assumes that realistic augmentations on the input data will not change the output distribution. Psuedo-labelling generates labels for unlabelled data before

Fig. 1: Overview of our two complementary loss functions. The Action Affinity loss imposes the best matched (denoted by a check mark) prior of action compositions and distributions from the labelled data. The Action Continuity Loss removes the fragments of action labels.

training. These techniques are not suitable for video-based tasks for several reasons. Firstly, it is non-trivial to perform realistic data augmentation operations required by consistency regularization methods as the prevailing practice in temporal action segmentation is to use pre-computed feature vectors as input. Furthermore, directly extending naïve pseudo-labelling on videos may result in confirmation bias [1], *i.e.*, overfitting to incorrect pseudo-labels.

Constructing conceivable supervision for unlabelled data in temporal action segmentation task raises questions such as *"What action compositions are likely to occur?", "What is a reasonable temporal proportion for each action to take?"* and *"What kind of constraints should the action labels follow?"* We propose to tackle these questions by leveraging two unique observations we made on procedural videos: 1) **Action Affinity**, procedural videos performing a specific activity (*e.g.*, 'making_coffee') comprise correlated action units (*e.g.*, 'take_cup', 'pour_coffee', 'pour_milk', and 'stir_coffee') and there exist pairs of videos that have resembling temporal portions for action unit pairs; 2) **Action Continuity**, action labels stay locally constant and action labels only transit at true boundaries. The former is an observation on relations between video instances, while the latter is a video-wise trait. In this work, we propose two novel unsupervised loss functions, action affinity loss and action continuity loss, each leveraging one of these observations. An overview of our losses is depicted in Fig. 1.

As opposed to previous work [21], which uses sparse per-segment labelled frames for *every* video, we use datasets that consist of a small set of densely labelled videos and a large set of unlabelled videos. Dense labels do more than just provide frame-level action labels in the video. They can also be used to establish prior information about the action compositions and distributions. We thus define for a labelled video a high-level representation – its action frequency, *i.e.*, the temporal proportion of actions. For unlabelled video without action information, we adopt a soft version based on the network predictions. Action frequency naturally indicates the relative action lengths as it is normalised by video lengths. The fact that it does not constrain the action ordering allows for flexibility as some actions do not necessarily follow a rigid sequence. Considering the action frequency of labelled videos as action priors, we exploit the action

affinities between labelled and unlabelled video samples in a heuristic way and integrate them in the action affinity loss function for model training.

Fragmentation or over-segmentation is a common problem in action segmentation [13], which is exacerbated in a semi-supervised setting when networks train and overfit on few labelled samples. To mitigate this, we first propose an action sequence extraction scheme that better captures the underlying action ordering. These sequentially sub-sampled actions are later compared against the original network predictions with dynamic time warping [24] to estimate our action continuity loss. We show that the action continuity loss can take the same form as the classification cross-entropy loss with a proper distance function adopted in dynamic time warping. Although our loss function also aims to maintain the temporal continuity of actions, it differs from the commonly used agnostic smoothing loss of [8]. Ours enforces a specific action ordering while providing frame-wise action supervision.

To add robustness to the action boundaries found by dynamic time warping for the unlabelled videos, we propose a soft transitional boundary. Specifically, we smooth the rigid boundaries so that frames around the boundaries have a mixed probability of belonging to both consecutive action classes. This was previously explored in a weakly-supervised setting [6], albeit in a highly rigid form. In our work, we vary the number of boundary frames depending on the action duration and use a sigmoid function for mixing.

In summary, this paper offers four key contributions:

1. By investigating the correlation of actions between labelled and unlabelled procedural videos, we propose an action affinity loss to integrate action priors for semi-supervised learning.
2. Building on the continuity property of procedural actions, we propose an action continuity loss to enforce action ordering constraints and provide classification supervision for unlabelled data.
3. For more robust and reliable learning, we propose a general adaptive boundary smoothing (ABS) technique that generates smoothed coarse action probabilities for boundary frames. Our ABS improves segmentation performance in both semi- and fully-supervised settings.
4. Experimental results show that our proposed approach improves the segmentation performance by a large margin with a small amount of labelled data (5% and 10%) and achieve comparable performance to the fully-supervised setting with 50% of labelled data.

2 Related Work

2.1 Temporal Action Segmentation

Temporal action segmentation in videos has been explored with various levels of supervision. **Fully-supervised** approaches require the long videos in the training set to be densely annotated. The input and label pairs are fed into a Temporal Convolution Network (TCN) to learn a mapping to frame-wise action labels [17,

19, 8, 28]. **Weakly-supervised** methods use action lists [15] or action sets [22] to learn the alignment between actions and frames. Specifically, D3TW [2] proposed a differentiable dynamic time warping loss with continuous relaxation to discriminatively learn the best alignment when multiple action lists were provided. Compared to their work, we do not use any action list supervision, and we utilise DTW as an optimisation tool for making the best assignment that meets the ordering constraints. A more recent weakly-supervised work [21] learns to segment actions via a small percentage of action timestamps. Although weak supervision reduces the effort in frame-wise action labelling, it is still necessary to provide supervision for *every* video. **Unsupervised** approaches address action segmentation by combining clustering methods with temporal models (*e.g.*, Hidden Markov Model) [27, 16, 20]. While some work simply perform clustering on the input features which do not involve any learning and achieve very competitive results [26, 7]. Since no semantic labels are provided during learning, performances are evaluated based on the best Hungarian matching scores. One recent work ICC [29] proposed a contrastive learning approach for unsupervised learning of frame-wise features. The learned representations are then adapted to a semi-supervised setting by learning a post-hoc linear classifier. The classifier incorporates the unlabelled data with naïve pseudo-labels, which weakens the overall contribution to the semi-supervised learning area.

2.2 Semi-supervised Learning

Existing dominant approaches to image-based semi-supervised learning include consistency regularization [25, 31] and pseudo-labelling [18]. Consistency regularization methods, such as Temporal Ensembling [25] and Mean-teacher [31], aim to learn the prediction consistency in different epochs or models with augmented inputs. Applying augmentations analogous to the image domain such as flipping, rotation, and transformation to videos for action segmentation is non-trivial as the inputs are pre-computed feature vectors.

Pseudo-labelling methods generate labels for unlabelled data to guide learning [18]. To generate pseudo-labels, [5] leverages the sample similarity in the feature space to assign soft labels, whereas [12] implements a graph-based label propagation framework. Some researchers [4, 32] have attempted to apply semi-supervised learning to video tasks by adapting image-based techniques to take video input. However, little has been done to evaluate the effectiveness of semi-supervised learning in temporal action segmentation. To address this research gap, we propose two novel loss functions designed based on the observation of two unique properties of procedural task videos.

3 Method

3.1 Preliminaries

We denote a labelled sample video sequence of temporal length T as $\{(x^t, y^t)\}_{t=1}^{T}$, where x^t is the video frame feature indexed at time t and y^t is its semantic

action label. In a semi-supervised scenario, a labelled set $\mathcal{D}_L = \{(x_i, y_i)\}_{i=1}^{N}$ of N labelled videos and an unlabelled set $\mathcal{D}_U = \{(x_j)\}_{j=1}^{M}$ of M videos are given, where $M \gg N$. For every video in the small labelled set $\mathcal{D}_L$, each frame has a label from one of K classes, *i.e.*, $y_i^t \in \{1, 2, \ldots, K\}$. The complete training set is denoted by $\mathcal{D} = \mathcal{D}_L + \mathcal{D}_U$.

To learn a semi-supervised action segmentation model $\mathcal{M}$ parameterised by θ, we use the labelled and unlabelled videos with the following objective:

$$\min_{\theta} \sum_{(x,y)\in\mathcal{D}_L} \mathcal{L}^L(x, y; \theta) + \alpha \sum_{(x)\in\mathcal{D}_U} \mathcal{L}^U(x; \theta) + \beta \sum_{(x)\in\mathcal{D}} \mathcal{R}^D(x; \theta), \tag{1}$$

where $\mathcal{L}^L$ denotes a supervised loss (Sec. 3.2), $\mathcal{L}^U$ denotes an unsupervised loss, and $\mathcal{R}^D$ is some regularization loss (Sec. 3.2), weighted by hyperparameters $\alpha, \beta \in \mathbb{R}_{>0}$. In the above objective, formulating the unsupervised loss $\mathcal{L}_{\text{u}}$ is vital for effective semi-supervised learning. In this work, we designed two novel loss functions for unlabelled data, *i.e.* action affinity loss (Sec. 3.3) and action continuity loss (Sec. 3.4), each attending to one characteristic we observed in procedural videos.

3.2 Supervised Temporal Action Segmentation

Existing supervised approaches [8, 17, 28] take sequences of video frames as input and predict frame-wise action labels as a classification task. For the labelled data, we follow the same scheme to train the segmentation model $\mathcal{M}$ that estimates frame-wise action probabilities $p^t(k) = \mathcal{M}(x^t)$ with the classification loss formulated as:

$$\mathcal{L}_{\text{cls}}^L = \frac{1}{T} \sum_t -\log(p^t(y^t)) \tag{2}$$

where $p^t \in \mathbb{R}^K$ is the estimated action class probability for frame x^t. It is also common to apply a smoothing loss with threshold τ to encourage smooth transitions between frames:

$$\mathcal{L}_{\text{sm}} = \frac{1}{TK} \sum_{t,k} \tilde{\Delta}_{t,k}^2, \quad \tilde{\Delta}_{t,k} = \begin{cases} \Delta_{t,k} & : \Delta_{t,k} \leq \tau \\ \tau & : \text{otherwise} \end{cases}, \tag{3}$$

$$\Delta_{t,k} = \left| \log p^t(k) - \log p^{t-1}(k) \right|. \tag{4}$$

We follow [8] and set τ to 4.

3.3 Action Affinity

Videos performing the same (procedural) activity will share the same or a similar set of composing actions. We assume we can find similar videos that match and share resembling temporal proportions. With this motivation in mind, we define for a labelled video i a video-level representation based on the action frequency:

$$q_i(k) = \frac{1}{T_i} \sum_t^{T_i} \mathbb{1}(y_i^t == k); \quad k \in [1, \ldots, K] \tag{5}$$

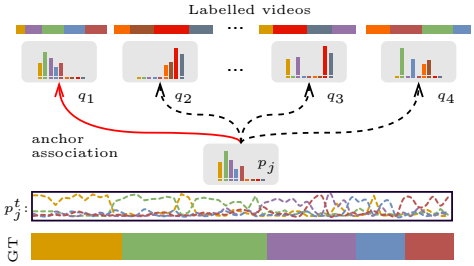


Fig. 2: Action affinity loss overview. Action frequencies q, p are first built for both labelled and unlabelled videos. The action affinity loss associates (red arrow) for p_j its nearest anchor q_1 in labelled set and then imposes the action prior from q_1 on p_j to supervise the learning. Our affinity loss allows variations of action ordering (green and purple segments in GT and q_1).

For an unlabelled video j, we define a soft action frequency based on the network prediction outputs:

$$p_j(k) = \frac{1}{T_j} \sum_{t}^{T_j} p_j^t(k); \quad k \in [1, \dots, K]. \tag{6}$$

Unless explicitly stated otherwise, we denote q, indexed by i, for labelled videos and p with index j for unlabelled videos.

Anchor Association. We want to provide action-level supervision for unlabelled videos by finding their most similar peers from the labelled set. Given some distance function $d(\cdot)$, we refer to a labelled video i as an anchor a_j for a unlabelled video j if their action frequencies are the closest amongst the entire labelled set, *i.e.*

$$a_j = q_{i^*}, \quad i^* = \arg\min_i d(q_i, p_j) \tag{7}$$

Affinity Loss. Formally, we use the Kullback–Leibler (KL) divergence as the distance criterion and define our action affinity loss as the affinity between the best matched pairs (p_j, a_j), which is also the minimum distance between p over the entire labelled set:

$$\mathcal{L}_{\text{aff}} = \sum_k a_j(k) \log \left(\frac{a_j(k)}{p_j(k)} \right) = \min_i \sum_k q_i(k) \log \left(\frac{q_i(k)}{p_j(k)} \right) \tag{8}$$

Minimising the above action affinity loss imposes pair-wise action frequency prior from the labelled set; it guides network outputs to have similar action composition to labelled videos, which is especially important when using unlabelled sequences for training. Fig. 2 depicts our affinity loss. Empirically, this loss combined with a frame-wise entropy loss outperforms pseudo-labels (see. Sec. 4.4).

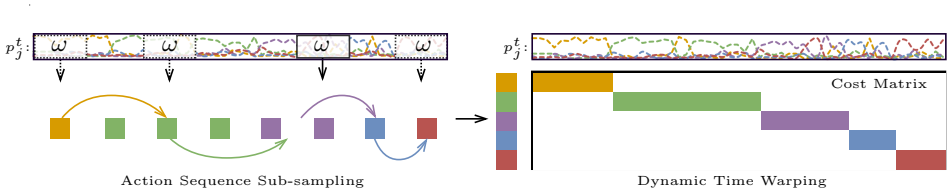


Fig. 3: Action continuity loss overview. Given network predictions for an unlabeled video p_j, a sliding window sub-sampling is first performed to obtain an action sequence (order indicated by coloured arrows); the sequence is later compared against p_j to construct a cost matrix. The action continuity loss is the average cost along the optimal assignment path (coloured segments in the cost matrix) found via dynamic time warping.

3.4 Action Continuity

A simple way to generate pseudo-labels $\hat{y}$ for unlabelled video sequences is to use the class label with the maximum probability, which can be used to supervise the learning of unlabelled data with the classification loss:

$$\mathcal{L}_{\text{pse}} = -\frac{1}{T}\sum_{t}\log(p^t(\hat{y}^t)), \text{ where } \hat{y}^t = \arg\max_k p^t(k). \tag{9}$$

However, such naïve pseudo-labels directly inferred from network outputs tend to be temporally over-fragmented [13]. This breaks the temporal continuity of actions, *i.e.*, label changes should occur only at (true) action boundaries. To this end, we propose an action continuity loss to impose such action transition constraints. For an unlabelled video, this loss takes as input its frame-wise predictions, sub-samples in time and then estimates the learning objective via dynamic time warping as illustrated in Fig. 3.

Action Sequence Sub-sampling. We first generate action candidates which have maximum average class probability within a sliding window of stride ω,

$$o = \arg\max_k \frac{1}{\omega}\sum_{t=t'}^{t'+\omega} p^t(k), \tag{10}$$

where t' is the previous temporal window location. Subsequently, we yield an ordered sequence with $\lceil \frac{T}{\omega} \rceil$ elements denoted as $\mathcal{O} = \{o^l\}_{l=1}^{\lceil \frac{T}{\omega} \rceil}$. This sequence can be further reduced in length by removing the adjacent action repetitions, *i.e.*, $o^l = o^{l+1}$, to a length of L.

Dynamic Time Warping. Given an unlabelled video with its frame-wise action probabilities p_j of length T and its inferred action sequence $\mathcal{O}$ of length L as described above, a cost matrix of alignment $\Delta = \{d(l,t)\} \in \mathbb{R}^{L\times T}$ can be constructed with some distance function $d(\cdot,\cdot)$. Using dynamic time warping, we find the best possible alignment Y^* defined by the following objective:

$$Y^* = \arg\min_Y \langle Y, \Delta \rangle \tag{11}$$

where $\langle \cdot, \cdot \rangle$ is the inner product and $Y \subset \{0,1\}^{L \times T}$ is a binary assignment matrix. $Y_{tl} = 1$ if frame t has the label o^l and $Y_{tl} = 0$ otherwise. Eq. (11) is solved efficiently with dynamic programming. The label assignment $\tilde{y}^t$ for p^t can be then inferred by parsing Y^*:

$$\tilde{y}^t = \sum_l o^l \mathbb{1}(Y^*_{lt} == 1) \tag{12}$$

Continuity Loss. An intuitive way of forming the continuity loss is to take the optimal objective of dynamic time warping and minimise it:

$$\mathcal{L}_{\text{cont}} = \langle Y^*, \Delta \rangle \tag{13}$$

We achieve this by choosing a specific distance function d. With a slight abuse of notation, we denote the categorical label o (Eq. (10)) as its one-hot embedding when written as $o(k)$ and designate the distance function as the KL divergence:

$$d(l,t) = KL(o^l || p^t) = \sum_k o^l(k) \log \left(\frac{o^l(k)}{p^t(k)} \right). \tag{14}$$

If we replace the o^l with the final assignment $\tilde{y}^t$ for p^t in Eq. (14), the cost for p^t in the optimal Y^* would become the negative log-likelihood $-\log(p^t(\tilde{y}^t))$. Averaging the cost over the entire video sequence leads to our final action continuity loss formulation:

$$\mathcal{L}_{\text{cont}} = \frac{1}{T} \min_Y \langle Y, \Delta \rangle = \frac{1}{T} \sum_t -\log(p^t(\tilde{y}^t)). \tag{15}$$

We note that with the KL divergence distance function, this continuity loss is consistent with the frame-wise classification loss enforcing the network predictions to approximate $\tilde{y}$, which is temporally continuous.

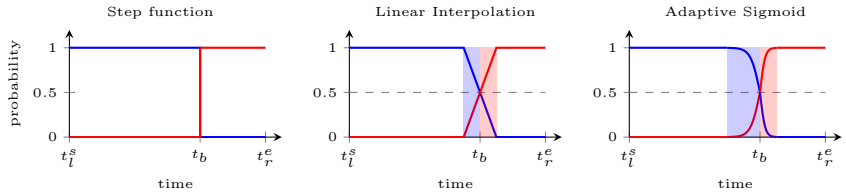
Adopting Eq. (2) for labelled data ($\mathcal{L}^L$), Eq. (8) and Eq. (15) for the unlabelled data ($\mathcal{L}^U$), and Eq. (3) for the regularization ($\mathcal{R}^D$), we can rewrite Eq. (1) as our semi-supervised learning objective with the following form:

$$\mathcal{L} = \mathcal{L}^L_{\text{cls}} + \alpha \mathcal{L}^U_{\text{aff}} + \beta \mathcal{L}^U_{\text{cont}} + \gamma \mathcal{L}^D_{\text{sm}} \tag{16}$$

where α, β, γ are trade-off parameters balancing the terms. The smoothing loss L^D_{sm} is imposed on the full set of data.

3.5 Adaptive Boundary Smoothing (ABS)

In our semi-supervised setting, the action boundaries of an unlabelled video inferred from the best possible assignment Y^* or $\tilde{y}$ may still be inaccurate. As such, we propose an adaptive boundary smoothing (ABS) technique to provide softer action boundary supervision for more robust and reliable learning. Boundary smoothing was initially proposed in [6] and has been explored in the weakly-supervised setting to improve the segmentation performance. Unlike [6],



(a) Standard One-hot Labels (b) Fixed-Duration Linear [6] (c) ABS (Ours)

Fig. 4: Probability assignment approaches around the action boundary as a function of time. Let t_b denote the estimated boundary between the left action in $[t_l^s, t_b)$ and the right action $[t_b, t_r^e)$. The blue and red shaded segments denote the boundary vicinities V_l and V_r. (a) The standard one-hot labels adopt a step function and assign hard action labels for all the frames. (b) The fixed-duration linear approach [6] mixes the action probabilities linearly with a fixed slope around the boundary. (c) ABS (Ours) uses a sigmoid function with a decay proportional to the action duration.

which uses a fixed linear-interpolation scheme for smoothing, we use an adaptive scheme based on the estimated action duration. This allows us to elastically mix action probabilities for frames within the vicinity of the boundary.

Duration Aware Boundary Vicinity. Given left and right action segments $(\mathcal{S}_l : [t_l^s, t_l^e, y_l], \mathcal{S}_r : [t_r^s, t_r^e, y_r])$ consecutive in time, let t^s, t^e denote the starting and ending timestamps of the action and $y \in [1, \ldots, K]$ the corresponding semantic label. The action boundary in between can be denoted as $t_b = t_r^s = t_l^e + 1$. With a vicinity parameter $v \in [0, 0.5]$, we define the boundary vincinities or ranges V_l and V_r for the left and right actions respectively:

$$V_l = [t_b - (t_b - t_l^s) * v, t_b), \quad \text{and} \quad V_r = [t_b, t_b + (t_r^e - t_b) * v). \tag{17}$$

Adaptive Sigmoid. Within each action boundary vicinity V, we utilize an adaptive sigmoid function to assign mixed probabilities. For a frame within the left boundary vicinity, *i.e.*, $t \in V_l$, its smoothed probabilities for two action classes (y_l, y_r) are written as:

$$y^t(y_l) = \frac{1}{1 + e^{-\frac{\epsilon}{|V_l|}(t - t_b)}}, \quad \text{and} \quad y^t(y_r) = 1 - y^t(y_l) \tag{18}$$

where ϵ is a predefined parameter which is set to 5 to ensure that the furthest frame to the boundary in a vicinity set has close to 1 probability for the action label of the segment it belongs to. $|V|$ denotes the temporal length of V. Probability assignment for V_r is identical to Eq. (18) but with y_r and y_l changed.

ABS can be efficiently incorporated in our approach by replacing the one-hot action probability within each boundary vicinity from $\tilde{y}$ (Eq. (12)) with the above mixed probabilities. With $v = 0$, our ABS degenerates into the one-hot setting. Fig. 4 compares three types of action probability assignments around the action boundaries. One-hot labels (Fig. 4(a)) are standard in practice and assume rigid action boundaries. Fixed-duration linear [6] (Fig. 4(b)) softens the

boundary with linearly interpolated action probabilities in a fix-sized temporal window. In contrast, our proposed ABS approach (Fig. 4(c)) allows the corresponding action probabilities of vicinity frames from a longer action segment to have a faster-descending speed when approaching the boundary and vice-versa. Smoothing in a larger vicinity of longer segments provides more training samples for shorter segments, while smoothing in a smaller vicinity of shorter segments helps preserve more high confident middle frames for learning.

4 Experiments

4.1 Datasets, Protocols and Evaluation

Datasets. We conducted our experiments on the three benchmark datasets. **Breakfast Actions** [14] comprises in total 1712 videos performing ten different activities with 48 actions. On average, each video contains six action instances. **50Salads** [30] has 50 videos with 17 action classes. **GTEA** [9] contains 28 videos of seven kitchen activities composing 11 different actions.

Protocols. We used the standard train-test splits for each dataset; we randomly selected 5%, 10% of the training set as the labelled set $\mathcal{D}_L$ and regarded the remaining training videos as the unlabelled set $\mathcal{D}_U$. The labelled set was ensured to contain at least one segment instance of each action. For 50Salads and GTEA, 3 and 5 videos were sampled in place of 5% and 10% of labelled data as the datasets are relatively small.

Evaluation. We adopted the same evaluation metrics as fully-supervised action segmentation and reported frame-wise accuracy (Acc), segmental edit score (Edit), and segmental F1 score with varying overlap thresholds 10%, 25%, and 50%. For all datasets, we randomly sampled five labelled subsets from the original training data. We cross-validated over the standard splits and reported the average over the splits across the five runs.

4.2 Implementation Details

We use the multi-stage temporal convolutional network (MS-TCN) [8] as the backbone segmentation model $\mathcal{M}$. Our model was first warmed up with only labelled data for 30 epochs, and then unlabelled data was incorporated for another 20 epochs. The initial learning rate was set as $5e^{-4}$. We used the Adam optimiser, with weights settings of $\alpha = 0.1$, $\beta = 0.01$, and $\gamma = 0.15$, as per [8]. The action sequence sub-sampling stride ω was set to 20. We set the vicinity parameter $v = 0.05$ for all three datasets.

4.3 Effectiveness

Table 1 reports the improvements of our method compared with the supervised baseline (*Base*) and naïve pseudo-labelling approach (*Pseudo*) on three benchmarks. *Base* is trained with only labelled data while *Pseudo* assigns pseudo-labels

Table 1: Performance of our proposed approach on three benchmark datasets

$\%D_L$	Method	Breakfast					50Salads					GTEA				
		F1@{10, 25, 50}			Edit	Acc	F1@{10, 25, 50}			Edit	Acc	F1@{10, 25, 50}			Edit	Acc
5	Base	36.7	28.4	19.5	37.5	28.2	26.8	19.7	11.5	26.1	28.1	29.9	25.8	14.8	31.0	37.2
	Pseudo	40.2	28.5	20.1	41.3	20.9	22.6	17.0	12.1	22.0	24.0	48.4	42.3	30.2	45.4	48.1
	Ours	44.5	35.3	26.5	45.9	38.1	37.4	32.3	25.5	32.9	52.3	59.8	53.6	39.0	55.7	55.8
	Gain	**7.8**	**6.9**	**7.0**	**8.4**	**9.9**	**10.6**	**12.6**	**14.0**	**6.8**	**24.2**	**29.9**	**27.8**	**24.2**	**24.7**	**18.6**
10	Base	46.8	41.1	29.2	50.9	37.1	27.6	24.3	16.0	27.4	32.0	38.1	29.6	15.3	39.6	41.1
	Pseudo	49.3	44.8	33.9	49.7	40.2	36.2	32.4	24.5	33.5	41.1	65.5	60.7	45.8	59.9	57.9
	Ours	56.9	51.3	39.0	57.7	49.5	47.3	42.7	31.8	43.6	58.0	71.5	66.0	52.9	67.2	62.6
	Gain	**10.1**	**10.2**	**9.8**	**6.8**	**12.4**	**19.7**	**18.4**	**15.8**	**16.2**	**26.0**	**33.4**	**36.4**	**37.6**	**27.6**	**21.5**

Table 2: Comparison of frame accuracy boost between different dataset variances

	labelled	50Salads	GTEA	Breakfast
var	-	$8e^{-4}$	$3e^{-3}$	$6e^{-3}$
Gain	5%	24.2	18.6	9.9
Gain	10%	26.0	21.5	12.4

Table 3: Effect of activity labels on Breakfast (5%)

	F1@{10,25,50}			Edit	Acc
w/o activity	44.5	35.3	26.5	45.9	38.1
w/ activity	56.6	49.3	35.8	59.4	56.6
Gain	**12.1**	**14.0**	**9.3**	**13.5**	**18.5**

and trains with $\mathcal{L}_{\text{pse}}$ (Eq. (9)). In both the 5% and 10% settings, our model consistently outperformed the *Base* model by a large margin. Specifically, on the 50Salads dataset, the accuracy of our model increased by 26% (from 32.0% $\rightarrow$ 58.0%). The overall increase in performance across datasets was greater when more labelled data (5%$\rightarrow$10%) was provided. It is noteworthy that on 50Salads with 5% labelled data, the segmentation performances for *Pseudo* are lower than the *Base* by around 3%, which shows that the model overfitted to inaccurate pseudo-labels, likely due to confirmation bias. On the contrary, our proposed approach can still significantly boost the accuracy performance by a large gain of 24.2%. This verifies the effectiveness of the valuable action affinity prior information inferred from the rarely few labelled video samples.

Affinity Association. Amongst all three datasets in Table 1, the increase in Acc performance was the greatest on 50Salads and the least on Breakfast. We speculate that this is related to how accurate the affinity association (Eq. (7)) is in finding the anchor videos from the labelled set. We validated this by calculating the total variance amongst the full set $\mathcal{D}$. The total variance is defined as the trace of the action frequency covariance matrix $\mathcal{C} \in \mathbb{R}^{K\times K}$ normalized by the number of actions, *i.e.*, $var = \text{tr}(\mathcal{C})/K$. The lower the variance, the more likely our affinity loss provided accurate supervision. The extreme case where the full set of videos share the identical action composition and frequency ($var = 0$) guarantees that the supervision by affinity loss is always accurate and precise. Table 2 verifies that datasets with smaller variances had higher accuracy gains. Breakfast has the largest variance because it has 10 activities with slightly overlapping composing actions. By extension, its dispersed action frequency representations would cause the variance for a single action to be high.

We also evaluated our approach on the Breakfast dataset with video-level activity labels provided for all videos and reported the results in Table. 3. In this setting, we only searched for anchors from labelled videos with the same activity label. As we can see, when activity labels were given, the performance had a striking improvement of 18.5% in accuracy. This is because these high-level labels excluded the incorrect anchor associations across two different activities. Such improvement validates our affinity observation in the same activity videos.

4.4 Ablation Studies

Loss Functions and ABS. Table 4 reports the ablation study results on different variants of loss functions and ABS. The first row is the baseline model trained with only labelled data and $\mathcal{L}_{\text{cls}}$. Results for naïve pseudo-labelling loss $\mathcal{L}_{\text{pse}}$ (Eq. (9)) in the second row show a mild increase in F1 scores and accuracy compared to the baseline. While more unlabelled data was accessible for learning, using them in the form of pseudo-labels brought little advantage. On the other hand, our proposed action affinity loss $\mathcal{L}_{\text{aff}}$ (third row) surpassed the pseudo-labelling counterpart by a margin of around 2% on all metrics. We imposed an extra frame-wise entropy loss formulated as $-\sum_k p_i^t(k) \log p_i^t(k)$ in this variant, which forced the network to produce confident frame-wise predictions as the affinity loss does not provide frame-level supervision. The combination of action affinity and naïve pseudo-labelling (fourth row) further enhanced the performance. Such improvements show that our affinity loss $\mathcal{L}_{\text{aff}}$ can improve the quality of pseudo-labels, which we will evaluate in the following text. Our model combining $\mathcal{L}_{\text{aff}}$ and $\mathcal{L}_{\text{cont}}$ achieved better performance than all the above variants. Lastly, as indicated by the last row, the integration of our proposed ABS further boosted the segmentation performance on Breakfast.

Table 4: Loss function ablation study on Breakfast (10%)

$\mathcal{L}_{\text{cls}}$	$\mathcal{L}_{\text{pse}}$	$\mathcal{L}_{\text{aff}}$	$\mathcal{L}_{\text{cont}}$	ABS	F1@{10, 25, 50}			Edit	Acc
✓					47.9	40.6	28.6	51.8	36.8
✓	✓				49.3	44.8	33.9	49.7	40.2
✓		✓			52.0	46.5	34.3	53.4	44.0
✓	✓	✓			54.1	46.7	34.9	54.1	47.8
✓		✓	✓		53.8	50.1	37.6	56.6	49.2
✓		✓	✓	✓	**56.9**	**51.3**	**39.0**	**57.7**	**49.5**

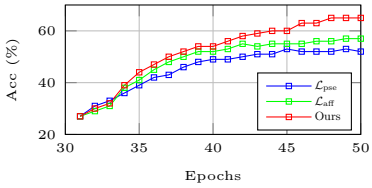


Fig. 5: Pseudo-label accuracy against training epochs on GTEA (10%).

Pseudo-labels. We studied the quality of estimated action classes on the unlabelled data $\mathcal{D}_U$. Fig. 5 shows a plot of the pseudo-label accuracy between training epochs. In the first epoch that unlabelled data was incorporated for training, all variants achieved the same accuracy, but the scores diverged as the training progressed. Imposing $\mathcal{L}_{\text{aff}}$ led to better accuracy compared to $\mathcal{L}_{\text{pse}}$, while our full loss formulation predicted the most accurate pseudo-labels for unlabelled data.

Table 5: Sub-sampling stride ω on Breakfast (10%)

ω	10	15	20	25	30	60
Acc	48.5	48.8	**49.5**	49.0	47.9	45.6

Table 6: Effect of α, β on GTEA (10%)

	α			
β	1	0.1	0.01	0.001
1	57.0	59.3	58.7	54.5
0.1	58.9	60.3	61.5	57.9
0.01	59.2	**62.6**	61.9	58.0
0.001	58.3	61.4	60.5	57.3

Table 7: Effectiveness of ABS for fully-supervised action segmentation

	Breakfast					50Salads					GTEA				
	F1@{10,25,50}			Edit	Acc	F1@{10,25,50}			Edit	Acc	F1@{10,25,50}			Edit	Acc
Base	63.2	57.7	45.6	65.5	65.1	66.8	63.7	55.2	59.8	78.2	84.9	82.4	67.6	79.7	76.6
+ABS	71.3	65.9	52.2	71.8	68.9	72.5	70.1	61.8	66.8	79.8	87.6	85.4	71.7	82.8	77.4
Gain	**8.1**	**8.2**	**6.6**	**6.3**	**3.8**	**5.7**	**6.4**	**6.6**	**7.0**	**1.6**	**2.7**	**3.0**	**4.1**	**3.1**	**0.8**

Sub-sampling Stride ω. Table 5 shows the accuracy changes with respect to the sub-sampling stride ω in Eq. (10) on Breakfast with 10% labelled data. The frame accuracy fluctuates around 49% with small strides, but when the stride becomes too large, *e.g.*, $\omega = 60$, the performance dropped by a margin of 3.9% as some actions are likely to be fully skipped during sub-sampling. The best accuracy of 49.5% was achieved when $\omega = 20$.

Loss Hyperparameters. The effect of hyperparameters is presented in Table 6. A very small weight on our affinity loss ($\alpha = 0.001$) led to the lowest performance, as indicated by the last column. Increasing α boosted the performance, which shows that the action priors from affinity loss is vital. The comparison between the rows indicates that a large weight for the action continuity loss, *e.g.*, $\beta = 1$, caused the model to overfit to the inaccurate pseudo-labels and produced inferior results since it also provided frame-wise pseudo-supervision. The overall best performance arrived at 62.6% with $\alpha = 0.1, \beta = 0.01$.

Vicinity Parameter v. Table 8 compares ABS against One-hot and Fixed-duration linear [6]. ASB with $v = 0.05$ enhanced the segmentation results by 1-2% compared to the baseline One-hot ($v = 0$). Fixed-duration linear [6] was also helpful, but the performance gain was only marginal. Setting $v = 0.1$ doubles the vicinity, which experienced a performance drop compared to $v = 0.05$; this likely indicates that the smoothing range is too large.

ABS for Supervised Learning. Given that ABS is a general smoothing technique, we further integrated ABS with the fully-supervised setting and report the results in Table 7. A consistent increase in segmentation performance compared to the baseline was observed across all three datasets. Also, the relatively large improvements were made in segmental metrics (F1 and Edit scores).

Table 8: Comparison of vicinity v on Breakfast (10%)

Method	F1@{10,25,50}			Edit	Acc
One-hot($v = 0$)	53.8	50.1	37.6	56.6	49.2
Fixed-duration [6]	54.7	50.5	38.1	56.9	49.1
$v = 0.05$	**56.9**	**51.3**	**39.0**	**57.7**	**49.5**
$v = 0.1$	55.1	50.9	37.9	57.0	48.9

Table 9: Accuracy performance comparison with approaches under various supervisions, * denotes test data used for training

	Method	Breakfast	50salads	GTEA
Full	MSTCN [8]	65.1	78.2	76.6
	SSTDA [3]*	**70.2**	**83.2**	79.8
	Ours (100%)*	69.3	82.5	**80.4**
Weak	Timestamp [21]	64.1	75.6	66.4
	SSTDA [3] (65%)*	65.8	80.7	75.7
Semi	Ours (5%)	38.1	52.3	55.8
	Ours (10%)	49.5	58.0	62.6
	Ours (50%)	**63.9**	**78.8**	**77.9**

4.5 Comparison to State-of-the-Art Approaches

We list in Table 9 relevant state-of-the-art approaches adopting MS-TCN [8] or its variants as the backbone for a fair comparison. We did not include ICC [29] as their approach cannot work with the MS-TCN architecture. For the "Full" comparison, we followed SSTDA [3] and applied our semi-supervised method to 100% labelled data. We used the test data as the unlabelled set and achieved comparable performance. The frame accuracy of Timestamp [21], which uses per-segment supervision for all video samples, is close to fully-supervised MS-TCN [8] except on GTEA. With 50% labelled data, our approach managed to achieve comparable or better performance compared to Timestamp [21], MS-TCN [8] as well as SSTDA [3] using a larger percentage (65%) of labelled data.

5 Conclusion

Procedural videos performing the same tasks exhibit affinity in action composition and continuity in action duration. Based on these unique characteristics, we proposed two novel loss functions for the semi-supervised temporal action segmentation task. The action affinity loss harnessed the action priors from the labelled set to supervise the unlabelled data. The action continuity loss function sub-sampled action sequence to enforce the temporal continuity of actions and provided frame-wise supervision. Furthermore, we proposed an adaptive boundary smoothing technique for more robust action boundaries. Our approach significantly improves the segmentation performance with a very small amount (5% and 10%) of labelled data and reaches comparable performance to the full supervision methods with 50% labelled videos.

Acknowledgements: This research is supported by the National Research Foundation, Singapore under its NRF Fellowship for AI (NRF-NRFFAI1-2019-0001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

1. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: IJCNN. pp. 1–8. IEEE (2020)
2. Chang, C.Y., Huang, D.A., Sui, Y., Fei-Fei, L., Niebles, J.C.: D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In: CVPR. pp. 3546–3555 (2019)
3. Chen, M.H., Li, B., Bao, Y., AlRegib, G., Kira, Z.: Action segmentation with joint self-supervised temporal domain adaptation. In: CVPR. pp. 9454–9463 (2020)
4. Cho, S., Lee, H., Kim, M., Jang, S., Lee, S.: Pixel-level bijective matching for video object segmentation. In: WACV. pp. 129–138 (2022)
5. Ding, G., Zhang, S., Khan, S., Tang, Z., Zhang, J., Porikli, F.: Feature affinity-based pseudo labeling for semi-supervised person re-identification. TMM **21**(11), 2891–2902 (2019)
6. Ding, L., Xu, C.: Weakly-supervised action segmentation with iterative soft boundary assignment. In: CVPR. pp. 6508–6516 (2018)
7. Du, Z., Wang, X., Zhou, G., Wang, Q.: 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)
8. Farha, Y.A., Gall, J.: Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: CVPR. pp. 3575–3584 (2019)
9. Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. In: CVPR (2011)
10. Fayyaz, M., Gall, J.: Sct: Set constrained temporal transformer for set supervised action segmentation. In: CVPR. pp. 501–510 (2020)
11. He, R., Yang, J., Qi, X.: Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In: ICCV. pp. 6930–6940 (2021)
12. Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Label propagation for deep semi-supervised learning. In: CVPR. pp. 5070–5079 (2019)
13. Ishikawa, Y., Kasai, S., Aoki, Y., Kataoka, H.: Alleviating over-segmentation errors by detecting action boundaries. In: WACV. pp. 2322–2331 (2021)
14. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: CVPR (2014)
15. Kuehne, H., Richard, A., Gall, J.: Weakly supervised learning of actions from transcripts. CVIU **163**, 78–89 (2017)
16. Kukleva, A., Kuehne, H., Sener, F., Gall, J.: Unsupervised learning of action classes with continuous temporal embedding. In: CVPR. pp. 12066–12074 (2019)
17. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: CVPR. pp. 156–165 (2017)
18. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML. vol. 3, p. 896 (2013)
19. Lei, P., Todorovic, S.: Temporal deformable residual networks for action segmentation in videos. In: CVPR. pp. 6742–6751 (2018)
20. Li, J., Todorovic, S.: Action shuffle alternating learning for unsupervised action segmentation. In: CVPR. pp. 12628–12636 (2021)
21. Li, Z., Abu Farha, Y., Gall, J.: Temporal action segmentation from timestamp supervision. In: CVPR. pp. 8365–8374 (2021)

22. Richard, A., Kuehne, H., Gall, J.: Action sets: Weakly supervised action segmentation without ordering constraints. In: CVPR. pp. 5987–5996 (2018)
23. Rizve, M.N., Duarte, K., Rawat, Y.S., Shah, M.: In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. arXiv preprint arXiv:2101.06329 (2021)
24. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **26**(1), 43–49 (1978)
25. Samuli, L., Timo, A.: Temporal ensembling for semi-supervised learning. In: ICLR. vol. 4, p. 6 (2017)
26. 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)
27. Sener, F., Yao, A.: Unsupervised learning and segmentation of complex activities from video. In: CVPR. pp. 8368–8376 (2018)
28. Singhania, D., Rahaman, R., Yao, A.: Coarse to fine multi-resolution temporal convolutional network. arXiv preprint arXiv:2105.10859 (2021)
29. Singhania, D., Rahaman, R., Yao, A.: Iterative contrast-classify for semi-supervised temporal action segmentation. arXiv preprint arXiv:2112.01402 (2021)
30. Stein, S., McKenna, S.J.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: ACM international joint conference on Pervasive and ubiquitous computing. pp. 729–738. ACM (2013)
31. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: NeurIPS. pp. 1195–1204 (2017)
32. Wang, X., Zhang, S., Qing, Z., Shao, Y., Gao, C., Sang, N.: Self-supervised learning for semi-supervised temporal action proposal. In: CVPR. pp. 1905–1914 (2021)
33. Wang, Z., Li, Y., Guo, Y., Fang, L., Wang, S.: Data-uncertainty guided multi-phase learning for semi-supervised object detection. In: CVPR. pp. 4568–4577 (2021)