Weakly Supervised Energy-Based Learning for Action Segmentation

Jun Li
Oregon State University
liju2@oregonstate.edu

Peng Lei∗
Amazon.com Services, Inc.
leipeng@amazon.com

Sinisa Todorovic
Oregon State University
sinisa@oregonstate.edu

Abstract

This paper is about labeling video frames with action classes under weak supervision in training, where we have access to a temporal ordering of actions, but their start and end frames in training videos are unknown. Following prior work, we use an HMM grounded on a Gated Recurrent Unit (GRU) for frame labeling. Our key contribution is a new constrained discriminative forward loss (CDFL) that we use for training the HMM and GRU under weak supervision. While prior work typically estimates the loss on a single, inferred video segmentation, our CDFL discriminates between the energy of all valid and invalid frame labelings of a training video. A valid frame labeling satisfies the ground-truth temporal ordering of actions, whereas an invalid one violates the ground truth. We specify an efficient recursive algorithm for computing the CDFL in terms of the logadd function of the segmentation energy. Our evaluation on action segmentation and alignment gives superior results to those of the state of the art on the benchmark Breakfast Action, Hollywood Extended, and 50Salads datasets.†

1. Introduction

This paper presents an approach to weakly supervised action segmentation by labeling video frames with action classes. Weak supervision means that in training our approach has access only to the temporal ordering of actions, but their ground-truth start and end frames are not provided. This is an important problem with a wide range of applications, since the more common fully supervised action segmentation typically requires expensive manual annotations of action occurrences in every video frame.

Our fundamental challenge is that the set of all possible segmentations of a training video may consist of multiple distinct valid segmentations that satisfy the provided ground-truth ordering of actions, along with invalid segmentations that violate the ground truth. It is not clear how to estimate loss (and subsequently train the segmenter) over multiple valid segmentations.

Motivation: Prior work [8, 12, 20, 7, 22] typically uses a temporal model (e.g., deep neural network, or HMM) to infer a single, valid, optimal video segmentation, and takes this inference result as a pseudo ground truth for estimating the incurred loss. However, a particular training video may exhibit a significant variation (not yet captured by the model along the course of training), which may negatively affect estimation of the pseudo ground truth, such that the inferred action segmentation is significantly different from the true one. In turn, the loss estimated on the incorrect pseudo ground truth may corrupt training by reducing, instead of maximizing, the discriminative margin between the ground truth and other valid segmentations. In this paper, we seek to alleviate these issues.

Contributions: Prior work shows that a statistical language model is useful for weakly supervised learning and modeling of video sequences [17, 9, 19, 22, 3]. Following [22], we also adopt a Hidden Markov Model (HMM) grounded on a Gated Recurrent Unit (GRU) [4] for labeling video frames. The major difference is that we do not generate a unique pseudo ground truth for training. Instead, we efficiently account for all candidate segmentations of a training video when estimating the loss. To this end, we formulate a new Constrained Discriminative Forward Loss (CDFL) as a difference between the energy of valid and invalid candidate video segmentations. In comparison with prior work, the CDFL improves robustness of our training, because minimizing the CDFL amounts to maximizing the discrimination margin between candidate segmentations that satisfy and violate ground truth, whereas prior work solely optimizes a score of the inferred single valid segmentation. Robustness of training is further improved when the CDFL takes into account only hard invalid segmentations whose edge energy is lower than that of valid ones. Along with the new CDFL formulation, our key contribution is a new recursive algorithm for efficiently estimating the CDFL in terms of the logadd function of the segmentation energy.

Our Approach: Fig. 1 shows an overview of our weakly supervised training of the HMM with GRU that consists of

†The code is available at https://github.com/JunLi-Galios/CDFL.
two steps. In the first step, we run a constrained Viterbi algorithm for HMM inference on a given training video so that the resulting segmentation is valid. This initial video segmentation is used for efficiently building a fully connected segmentation graph aimed at representing alternative candidate segmentations. In this graph, nodes represent segmentation cuts of the initially inferred segmentation – i.e., video frames where one action ends and a subsequent one starts – and edges represent video segments between every two temporally ordered cuts. For improving action boundary detection, we further augment the initial set of nodes with video frames that are in a vicinity of every cut, as well as the initial set of edges with corresponding temporal links between the added nodes. Directed paths of such a fully connected graph explicitly represent many candidate action segmentations, beyond the initial HMM’s inference.

The second step of our training efficiently computes a total energy score of frame labeling along all paths in the segmentation graph. Efficiency comes from our novel recursive estimation of the segmentation energy, where we exploit the accumulative property of the \( \logadd \) function. A difference of the accumulated energy of action labeling along the valid and invalid paths is used to compute the CDFL. In this paper, we also consider several other loss formulations expressed in terms of the energy of valid and invalid paths. The loss is then used for training HMM parameters and back-propagated to the GRU for end-to-end training.

For inference on a test video, as in the first step of our training, we use a constrained Viterbi algorithm to perform the HMM inference which will satisfy at least one action sequence seen in training. Then, we use this initial video segmentation as an anchor for building the segmentation graph that comprises paths with finer action boundaries. Our output is the MAP path in the graph.

For evaluation, we consider the tasks of action segmentation and action alignment, where the latter provides additional information on the temporal ordering of actions in the test video. For both tasks on the Breakfast Action dataset [10], Hollywood Extended dataset [1], and 50-Salads dataset [24], we outperform the state of the art.

In the following, Sec. 2 reviews related work, Sec. 3 formulates our HMM and Constrained Viterbi for action segmentation, Sec. 4 describes how we construct the segmentation graph, Sec. 5 specifies our CDFL and related loss functions, and Sec. 6 presents our evaluation.

2. Related Work

This section reviews closely related work on weakly supervised action segmentation and Graph Transformer Networks. While a review of fully supervised action segmentation [25, 14, 18, 16] is beyond our scope, it is worth mentioning that our approach uses the same recurrent deep models for frame labeling as in [23, 25, 6]. Also, our approach is motivated by [11, 19] which integrate HMMs and modeling of action length priors within a deep learning architecture.

Weakly supervised action segmentation has recently made much progress [24, 10, 20, 7, 22]. For example, Extended Connectionist Temporal Classification (ECTC) addresses action alignment under the constraint of being consistent with frame-to-frame visual similarity [8]. Also, action segmentation has been addressed with a convex relaxation of discriminative clustering, and efficiently solved with the conditional gradient (Frank-Wolfe) algorithm [1]. Other approaches use a local action model and a global temporal alignment model that are alternatively trained [12, 20]. Some methods initially predict a video segmentation with a temporal convolutional network, and then iteratively refine the action boundaries [7]. Other approaches first generate pseudo-ground-truth labels for all video frames, e.g., with the Viterbi algorithm [22], and then train a classifier on these frame labels by minimizing the standard cross entropy loss. Finally [21] addresses a different weakly supervised setting from ours when the ground truth provides only a set of actions present without their temporal ordering.

All these approaches base their learning and prediction on estimating a penalty or probability of labeling individ-
ual frames. In contrast, we use an energy-based framework with the following differences. First, in training, we minimize the total energy of valid paths in the segmentation graph rather than optimize labeling probabilities of each frame. Second, instead of considering a single optimal valid path in the segmentation graph, we specify a loss function in terms of all valid paths. Hence, the Viterbi-initialized training on pseudo-labels of frames [22] represents a special case of our training done only for one valid path. In addition, our loss enforces discriminative training by accounting for invalid paths in the segmentation graph. Unlike [3] that randomly selects invalid paths, we efficiently account for all hard invalid paths in training. Finally, our training is not iterative as in [12, 20], and does not require iterative refinement of action boundaries as in [7].

Our CDFL extends the loss used for training of the Graph Transformer Network (GTN) [15, 13, 2, 5]. To the best of our knowledge, the GTN has been used only for text parsing, and never for action segmentation. In comparison with the GTN training, we significantly reduce complexity by building the video’s segmentation graph. Also, while the loss used for training the GTN accounts for both valid and invalid text parses, it cannot handle the special case when valid parses have lower scores than invalid ones. In contrast, our CDFL effectively accounts for the energy of valid and invalid paths, even when valid paths have significantly lower energy than invalid paths in the segmentation graph.

3. Our Model for Action Segmentation

Problem Setup: For each training video of length \( T \), we are given unsupervised frame-level features, \( x_{1:T} = [x_1, x_2, \ldots, x_T] \), and the ground-truth ordering of action classes \( a_{1:N} = [a_1, a_2, \ldots, a_N] \), also referred to as the transcript. \( N \) is the length of the annotation sequence, and \( a_n \) is the \( n \)-th action class in \( a_{1:N} \) that belongs to the set of \( K \) action classes, \( a_n \in \mathcal{A} = \{1, 2, \ldots, K\} \). Note that \( T \) and \( N \) may vary across the training set, and that there may be more than one occurrences of the same action class spread out in \( a_{1:N} \) (but of course \( a_n \neq a_{n+1} \)).

In inference, given frame features \( x_{1:T} \) of a video, our goal is to find an optimal segmentation \((\hat{a}_{1:N}, \hat{l}_{1:N})\), where \( \hat{N} \) is the predicted length of the action sequence, and \( \hat{l}_{1:N} = [\hat{l}_1, \hat{l}_2, \ldots, \hat{l}_N] \) includes the predicted number of video frames \( \hat{l}_n \) occupied by the predicted action \( \hat{a}_n \).

The Model: We use an HMM to model the posterior distribution of a video segmentation \((a_{1:N}, l_{1:N})\) given \( x_{1:T} \) as

\[
p(a_{1:N}, l_{1:N} | x_{1:T}) \propto p(x_{1:T} | a_{1:N}, l_{1:N}) p(l_{1:N} | a_{1:N}) p(a_{1:N}),
\]

\[
= \left( \prod_{t=1}^{T} p(x_t | a_{n(t)}) \right) \left( \prod_{n=1}^{N} p(l_n | a_n) \right) p(a_{1:N}). \tag{1}
\]

In (1), the likelihood \( p(x_t | a) \) is estimated as

\[
p(x_t | a) \propto \frac{p(a | x_t)}{p(a)}, \tag{2}
\]

where \( p(a | x_t) \) is the GRU’s softmax score for action \( a \in \mathcal{A} \) at frame \( t \), and the prior distribution of action classes \( p(a) \) is an normalized frame frequency of action occurrences in the training dataset. The likelihood of action length is modeled as a class-dependent Poisson distribution

\[
p(l | a) = \frac{\lambda_a^l}{l!} e^{-\lambda_a}, \tag{3}
\]

where \( \lambda_a \) is the mean length for class \( a \in \mathcal{A} \). Finally, the joint prior \( p(a_{1:N}) \) is a constant if the transcript \( a_{1:N} \) exists in the training set; otherwise, \( p(a_{1:N}) = 0 \). The same modeling formulation was well-motivated and used in state of the art [22].

Constrained Viterbi Algorithm: Given a training video, we first find an optimal valid action segmentation \((\hat{a}_{1:N}, \hat{l}_{1:N})\) by maximizing (1) with a constrained Viterbi algorithm, which ensures that \( \hat{a}_{1:N} \) is equal to the annotated transcript, \( \hat{a}_{1:N} = a_{1:N} \). Similarly, for inference on a test video, we first perform the constrained Viterbi algorithm against all transcripts \( \{a_{1:N}\} \) seen in training, i.e., ensure that the predicted \( \hat{a}_{1:N} \) has been seen at least once in training. Thus, the initial step of our inference on a training or test video is the same as in [22].

Our key difference from [22], is that we use the initial \((\hat{a}_{1:N}, \hat{l}_{1:N})\) to efficiently build a fully connected segmentation graph of the video, as explained in Sec. 4. Importantly, in training, the segmentation graph is not constructed to find a more optimal video segmentation that improves upon the initial prediction. Instead, the graph is used to efficiently account for all valid and invalid segmentations.

Given a video \( x_{1:T} \) and a transcript \( a_{1:N} \), the constrained Viterbi algorithm recursively maximizes the posterior in (1) such that the first \( n \) action labels of the transcript \( a_{1:n} = [a_1, \ldots, a_n] \subseteq a_{1:N} \) are respected at time \( t \):

\[
p(a_{1:n}, \hat{l}_{1:n} | x_{1:t}) = \max_{t', t''} \left\{ p(a_{1:n-1}, \hat{l}_{1:n-1} | x_{1:t'}) \right\} \prod_{s=t}^{t'} p(x_s | a_n(s)) \prod_{n=1}^{N} p(l_n | a_n) p(a_{1:n}), \tag{4}
\]

where \( l_n = t - t' \). We set \( p(\cdot | x_{1:0}) = 1 \), and \( p(a_{1:n}) = \kappa \), where \( \kappa > 0 \) is a constant.

4. Constructing the Segmentation Graph

Given a video \( x_{1:T} \), we first run the constrained Viterbi algorithm to obtain an initial video segmentation \((\hat{a}_{1:N}, \hat{l}_{1:N})\). For simplicity, in the following, we ignore the symbol ". This initial segmentation is characterized by
A subset of valid paths $\mathcal{P}^V \subset \mathcal{P}$ satisfies the given transcript. The other paths are invalid, $\mathcal{P}^I = \mathcal{P} \setminus \mathcal{P}^V$.

In the next section, we explain how to efficiently compute a total energy score of the exponentially many paths in $\mathcal{P}$ for estimating our loss in training.

5. Constrained Discriminative Forward Loss

In this paper, we study three distinct loss functions, defined in terms of a total energy score of paths in $G$. As there are exponentially many paths in $G$, our key contribution is the algorithm for efficiently estimating their total energy. Below, we specify our three loss functions ordered by their complexity. As we will show in Sec. 6, we obtain the best performance when using the CDFL in training.

5.1. Forward Loss

We define a forward loss, $L_F$, in terms of a total energy of all valid paths using the standard $\logadd$ function as

$$L_F = -\log(\sum_{\pi \in \mathcal{P}^V} \exp(-E_\pi)), \tag{7}$$

where energy of a path $E_\pi$ is given by (6). As there are exponentially many paths in $\mathcal{P}^V$, we cannot directly compute $L_F$ as specified in (7). Therefore, we derive a novel recursive algorithm for accumulating the energy scores of edges along multiple paths, as specified below.

We begin by defining the $\logadd$ function as

$$\logadd(a, b) = -\log(\exp(-a) + \exp(-b)). \tag{8}$$

Note that the $\logadd$ function is commutative and associative, so it can be defined on a set $S$ in a recursive manner:

$$\logadd(S) = \logadd(S\setminus\{x\}, x), \tag{9}$$

where $x$ is an element in $S$. Therefore, the forward loss given by (7) can be expressed as

$$L_F = \logadd(\{E_\pi : \pi \in \mathcal{P}^V\}). \tag{10}$$

Below, we simplify notation as $L_F = \logadd(\mathcal{P}^V)$.

We recursively compute the energy score $\ell_i(a_{1:n})$ of a path that ends at node $i'$ and covers first $n$ labels of the ground truth $a_{1:n} = [a_1, ..., a_n] \subseteq a_{1:N}$ in terms of the $\logadd$ scores $\ell_i(a_{1:n-1})$ of all valid paths that end at node $i$, $i < i'$, and cover first $n - 1$ labels as

$$\ell_i(a_{1:n}) = \logadd(\{\ell_i(a_{1:n-1}) + w_{1:i'}(a_n) : i < i'\}). \tag{11}$$

To prove (11), suppose that

$$\ell_i(a_{1:n-1}) = \logadd(\{E_{\pi_i} : \pi_i \in \mathcal{P}^V\})$$

$$= -\log(\sum_{\pi_i \in \mathcal{P}^V} \exp(-E_{\pi_i})), \tag{12}$$
Then, we have
\[ \pi_{11} \]
the constraint sequence \( a \). For a training video with length \( T \), we define
\[ \pi \]
when we take only the initial segmentation cuts \( b_{1:N+1} \) as
nodes of \( G \) (i.e., the window size \( \Delta = 0 \)), the forward loss is equal to the training loss used in [22].

### 5.2. Discriminative Forward Loss

We also consider the Discriminative Forward Loss, \( L_{DF} \), which extends \( L_F \) by additionally accounting for invalid paths in \( G \):

\[ L_{DF} = \logadd(\mathcal{P}^V) - \alpha \logadd(\mathcal{P}), \]

where \( \logadd(\mathcal{P}) \) aggregates a total energy of all paths in \( G \), and \( \alpha > 0 \) is a regularization factor that controls the relative importance of the valid and invalid paths for \( L_{DF} \).

Alg. 2 summarizes our recursive algorithm for computing \( \logadd(\mathcal{P}) \) in (15), whereas Alg. 1 shows how to compute \( \logadd(\mathcal{P}^V) \) in (15).

One advantage of \( L_{DF} \) over \( L_F \) is that minimizing \( L_{DF} \) amounts to maximizing the decision margin between the valid and invalid paths. However, a potential shortcoming of \( L_{DF} \) is that valid paths might have little effect in (15). In the case, when the energy of valid paths dominates the total energy of all paths, the former gets effectively subtracted in (15), and hence has very little effect on learning.

Moreover, we observe that in some cases the backpropagation of \( L_{DF} \) is dominated by the invalid paths. This can be clearly seen from the following derivation. We compute the gradient \( \nabla L_{DF} \) as

\[ \nabla L_{DF} = \nabla \logadd(\mathcal{P}^V) - \alpha \nabla \logadd(\mathcal{P}), \]

\[ = c_1 \sum_{\pi \in \mathcal{P}^V} \exp(-E_\pi) \nabla E_\pi - c_2 \sum_{\pi \in \mathcal{P}} \exp(-E_\pi) \nabla E_\pi, \]

where

\[ c_1 = \frac{(1-\alpha) \sum_{\pi \in \mathcal{P}^V} \exp(-E_\pi) + \sum_{\pi \in \mathcal{P}} \exp(-E_\pi)}{\sum_{\pi \in \mathcal{P}} \exp(-E_\pi)}, \]

\[ c_2 = \frac{\sum_{\pi \in \mathcal{P}} \exp(-E_\pi)}{\sum_{\pi \in \mathcal{P}} \exp(-E_\pi)}. \]

From (16)–(17), we note that in the case of \( \alpha \rightarrow 1 \), the backpropagation will be dominated by the invalid paths, whereas there would be no effect for invalid paths in training if \( \alpha = 0 \). Sec. 6 presents how different choices of \( \alpha \) affect our performance.

In the next section, we define the constrained discriminative forward loss to address this issue.

### 5.3. Constrained Discriminative Forward Loss

We define the CDFL as

\[ L_{CDF} = \logadd(\mathcal{P}^V) - \logadd(\mathcal{P}^{I^c}), \]

where \( \mathcal{P}^{I^c} \) consists of a subset of invalid paths in \( G \), where each edge \( e_{ii'} \) gets assigned an action class \( a \) such that its weight \( w_{ii'}(a) \) is pseudo ground truth class for \( e_{ii'} \). This constraint effectively addresses the aforementioned issue when the valid paths have significantly lower energy than the invalid paths. Alg. 3 summarizes our recursive algorithm for computing
logadd(\(P^{i_k}\)) in (18), whereas Alg. 1 shows how to compute logadd(\(P^{i'}\)) in (18).

As \(L_{DF}\) accounts for the invalid paths, \(L_{CDF}\) further accounts for the hard invalid paths. Therefore, the model robustness is further improved by minimizing \(L_{CDF}\) which amounts to maximizing the decision margin between the valid and hard invalid paths.

5.4. Our Computational Efficiency

As summarized in Alg. 1–3, our training first runs the constrained Viterbi algorithm (see Sec. 3) to get the initial segmentation cuts with complexity \(O(T^2N)\) for a video of length \(T\) and the ground-truth action sequence of length \(N\). Then, CDFL efficiently accumulates the energy of both valid and invalid paths in \(G\) with complexity \(O(\Delta^2KN)\) for the neighborhood window size \(\Delta\) and the class set size \(K\). Therefore, our total complexity of training is \(O(T^2N + \Delta^2KN)\).

Note that prior work [22] also runs the Constrained Viterbi with complexity \(O(T^2N)\), so relative to theirs our complexity is increased by \(O(\Delta^2KN)\). This additional complexity is significantly smaller than \(O(T^2N)\) as \(\Delta^2K \ll T^2\). In our experimental evaluation, we get the best results for \(\Delta \leq 20\) frames, whereas video length \(T\) can go to several minutes.

6. Results

Both action segmentation and alignment are evaluated on the Breakfast Actions [10], Hollywood Extended [1], and 50Salads [24] datasets. We perform the same cross-validation strategy as the state of the art, and report our average results. We call our approach CDFL, trained with loss given by (18).

Datasets. For all datasets, we use as input the pre-processed, public, unsupervised frame-level features. The same frame features are used by [8, 12, 20, 22]. The features are dense trajectories represented by PCA-projected Fisher vectors [11]. Breakfast [10] consists of 1, 712 videos of people making breakfast with 10 cooking activities. The cooking activities are comprised of 48 action classes. On average, every video has 6.9 action instances, and the video length ranges from a few seconds to several minutes. Hollywood Extended [1] contains 937 video clips from different Hollywood movies, showing 16 action classes. Each clip contains 2.5 actions on average. 50Salads [24] has 50 very long videos showing 17 classes of human manipulative gestures. On average, each video has 20 action instances. There are 600,000 annotated frames.

Evaluation Metrics. We use the following four standard metrics, as in [1, 7]. The mean-over-frames (Mof) is the average percentage of correctly labeled frames. To overcome the potential drawback that frames are dominated by background class, we compute mean-over-frames without background (Mof-bg) as the average percentage of correctly labeled video frames with background frames removed.

<table>
<thead>
<tr>
<th>Breakfast</th>
<th>Mof</th>
<th>Mof-bg</th>
<th>IoU</th>
<th>IoD</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCDC[1]</td>
<td>8.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CTC[8]</td>
<td>21.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HTK [11]</td>
<td>25.9</td>
<td>9.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ECTC [8]</td>
<td>27.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HMM/RNN [20]</td>
<td>33.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TCFPN [7]</td>
<td>38.4</td>
<td>24.2</td>
<td>40.6</td>
<td>-</td>
</tr>
<tr>
<td>NN-Viterbi [22]</td>
<td>43.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>D3TW [3]</td>
<td>45.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Our CDFL</td>
<td>50.2</td>
<td>48.0</td>
<td>33.7</td>
<td>45.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hollywood Ext</th>
<th>Mof</th>
<th>Mof-bg</th>
<th>IoU</th>
<th>IoD</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTK [11]</td>
<td>33.0</td>
<td>8.6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HMM/RNN [20]</td>
<td>-</td>
<td>11.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TCFPN [7]</td>
<td>28.7</td>
<td>34.5</td>
<td>12.6</td>
<td>18.3</td>
</tr>
<tr>
<td>D3TW [3]</td>
<td>33.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Our CDFL</td>
<td>45.0</td>
<td>40.6</td>
<td>19.5</td>
<td>25.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>50Salads</th>
<th>Mof</th>
<th>Mof-bg</th>
<th>IoU</th>
<th>IoD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTC[8]</td>
<td>11.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HTK [11]</td>
<td>24.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HMM/RNN [20]</td>
<td>45.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NN-Viterbi [22]</td>
<td>49.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Our CDFL</td>
<td>54.7</td>
<td>49.8</td>
<td>31.5</td>
<td>40.4</td>
</tr>
</tbody>
</table>

Table 1. Action segmentation evaluations on Breakfast, Hollywood Ext and 50Salads. The dash means no results reported by prior work.
The intersection over union (IoU) and the intersection over detection (IoD) are computed as $\text{IoU} = \frac{|GT \cap D|}{|GT \cup D|}$, and $\text{IoD} = \frac{|GT \cap D|}{|D|}$, where $|GT|$ denotes the extent of the ground truth segment and $|D|$ is the extent of a correctly detected action segment.

**Training.** We train a single-layer GRU with 64 hidden units in $10^5$ iterations, where for each iteration one training video is randomly selected. The initial learning rate of 0.01 is decreased to 0.001 at the 60,000th iteration. The mean action lengths $\lambda_\alpha$ in (3), and the action priors $p(\alpha)$ in (2) are estimated from the history of pseudo ground truths. Unlike [22], we do not use the history of pseudo ground truths for computing loss in the current iteration. Consequently, our training time per iteration is less than that of [22].

### 6.1. Action Segmentation

Tab. 1 compares CDFL with the state of the art. From the table, CDFL achieves the best performance in terms of all the four metrics. Fig. 3 qualitatively compares the ground truth and CDFL’s output on an example test video in Breakfast dataset. As can be seen, CDFL typically misses the true start or end of actions by only a few frames. In general, CDFL successfully detects most action occurrences.

**Ablation Study for Action Segmentation.** Tab. 2 compares our action segmentation performance on Breakfast when using different sizes of the neighborhood window placed around the initial segmentation cuts (as explained in Sec. 4) and different loss functions (as specified in Sec. 5). From the table, training by accounting for invalid paths in $L_{DF}$ and $L_{CDF}$ gives better performance than only accounting for valid paths in $L_F$. In addition, considering neighboring frames for action boundary refinement within a window around the initial segmentation cuts gives better performance than taking into account only a single optimal path in the segmentation graph when the window size is 0. The best test performance is achieved using $L_{CDF}$ with window size of 20 in training.

Fig. 4 illustrates the CDFL’s action segmentations on a sample test video $P03\_stereo01\_P03\_milk$ from Breakfast dataset using $L_{CDF}$, $L_{DF}$ and $L_F$, respectively. The background frames are marked in white. The result for $L_{CDF}$ is the best.

![Figure 3. Ground truth action sequence (take_cup, spoon_powder, pour_milk, stir_milk) (top) and our CDFL’s action segmentation (bottom) on the sample test video $P03\_stereo01\_P03\_milk$ from Breakfast dataset. The background frames are marked in white. CDFL may miss the true start and end of some actions, but successfully detects the actions.](image)

<table>
<thead>
<tr>
<th>Window Size</th>
<th>$L_F$</th>
<th>$L_{DF}$</th>
<th>$L_{CDF}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>43.5</td>
<td>40.9</td>
<td>46.6</td>
</tr>
<tr>
<td>20</td>
<td>44.3</td>
<td>40.9</td>
<td><strong>50.2</strong></td>
</tr>
<tr>
<td>10</td>
<td>43.8</td>
<td>40.0</td>
<td>46.2</td>
</tr>
<tr>
<td>0</td>
<td>43.0</td>
<td>38.7</td>
<td>45.0</td>
</tr>
</tbody>
</table>

Table 2. Mof and IoD evaluations on Breakfast for different neighborhood window sizes and different losses. CDFL with neighborhood size of 20 shows the best result.

![Figure 4. Top-down, the rows correspond to ground truth sequence of actions (pour_oil, crack_egg, fry_egg, put_egg2plate) and our action segmentations with neighbor-window size of 20 on the sample test video $P03\_cam01\_P03\_friedegg$ from Breakfast dataset using $L_{CDF}$, $L_{DF}$ and $L_F$, respectively. The background frames are marked in white. The result for $L_{CDF}$ is the best.](image)

![Figure 5. Ground truth action sequence (pour_oil, crack_egg, fry_egg, take_plate, put_egg2plate) (top) and CDFL’s action segmentations using different neighbor-window sizes on the sample test video $P04\_webcam02\_P04\_friedegg$ from Breakfast. The background frames are marked in white. The window size of 20 gives the best performance.](image)

![Image](image)

![Image](image)

6249
## 6. Results

<table>
<thead>
<tr>
<th>Breakfast</th>
<th>Mof</th>
<th>Mof-bg</th>
<th>IoU</th>
<th>IoD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECTC[8]</td>
<td>35.0</td>
<td>-</td>
<td>-</td>
<td>45.0</td>
</tr>
<tr>
<td>HTK [11]</td>
<td>43.9</td>
<td>-</td>
<td>26.6</td>
<td>42.6</td>
</tr>
<tr>
<td>OCDC [1]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>23.4</td>
</tr>
<tr>
<td>HMM/RNN [20]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>47.3</td>
</tr>
<tr>
<td>TCFPN [7]</td>
<td>53.5</td>
<td>51.7</td>
<td>35.3</td>
<td>52.3</td>
</tr>
<tr>
<td>ATDP [3]</td>
<td>57.0</td>
<td>-</td>
<td>-</td>
<td>56.3</td>
</tr>
<tr>
<td>Our CDFL</td>
<td><strong>63.0</strong></td>
<td><strong>61.4</strong></td>
<td><strong>45.8</strong></td>
<td><strong>63.9</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hollywood Ext</th>
<th>Mof</th>
<th>Mof-bg</th>
<th>IoU</th>
<th>IoD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECTC[8]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>41.0</td>
</tr>
<tr>
<td>HTK [11]</td>
<td>49.4</td>
<td>-</td>
<td>29.1</td>
<td>46.9</td>
</tr>
<tr>
<td>OCDC [1]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>43.9</td>
</tr>
<tr>
<td>HMM/RNN [20]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>46.3</td>
</tr>
<tr>
<td>TCFPN [7]</td>
<td>57.4</td>
<td>36.1</td>
<td>22.3</td>
<td>39.6</td>
</tr>
<tr>
<td>NN-Viterbi [22]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>48.7</td>
</tr>
<tr>
<td>ATDP [3]</td>
<td>59.4</td>
<td>-</td>
<td>-</td>
<td>50.9</td>
</tr>
<tr>
<td>Our CDFL</td>
<td><strong>64.3</strong></td>
<td><strong>70.8</strong></td>
<td><strong>40.5</strong></td>
<td><strong>52.9</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>50Salads</th>
<th>Mof</th>
<th>Mof-bg</th>
<th>IoU</th>
<th>IoD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our CDFL</td>
<td><strong>68.0</strong></td>
<td><strong>65.3</strong></td>
<td><strong>45.5</strong></td>
<td><strong>58.7</strong></td>
</tr>
</tbody>
</table>

Table 4. Action alignment evaluations on Breakfast, Hollywood Ext and 50Salads. The dash indicates that no results reported by prior work.

<table>
<thead>
<tr>
<th>Window size</th>
<th>$L_F$</th>
<th>$L_{DF}$</th>
<th>$L_{CDF}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>48.7</td>
<td>49.8</td>
<td>51.6</td>
</tr>
<tr>
<td>6</td>
<td>49.3</td>
<td>50.5</td>
<td><strong>52.9</strong></td>
</tr>
<tr>
<td>4</td>
<td>49.0</td>
<td>50.0</td>
<td>52.0</td>
</tr>
<tr>
<td>2</td>
<td>48.5</td>
<td>49.5</td>
<td>50.7</td>
</tr>
<tr>
<td>0</td>
<td>48.7</td>
<td>49.5</td>
<td>49.8</td>
</tr>
</tbody>
</table>

Table 5. IoD evaluations of our approach in action alignment on Hollywood Extended using different loss functions and different neighbor-window sizes in training. Using CDFL with neighbor-window size of 6 shows the best result.

![Figure 6. Ground truth action sequence (StandUp, SitDown, DriveCar, OpenDoor, OpenDoor, HugPerson) (top) and our action alignments (bottom) on the sample test video 0261 from Hollywood Extended. The background frames are marked in white. CDFL typically achieves a good action alignment.](image)

![Figure 7. Ground truth action sequence (OpenDoor, OpenDoor, OpenCarDoor) (top) and CDFL’s action alignments on the sample test video 0361 from Hollywood Extended, when trained using varying window sizes. The background frames are marked in white. Using CDFL and neighbor-window size of 6 gives the best results.](image)

## 7. Conclusion

We have extended the existing work on weakly supervised action segmentation that uses an HMM and GRU for labeling video frames by formulating a new energy-based learning on a video’s segmentation graph. The graph is constructed so as to facilitate computation of loss, expressed in terms of the energy of valid and invalid paths representing candidate action segmentations. Our key contribution is the new recursive algorithm for efficiently computing the accumulated energy of exponentially many paths in the segmentation graph. Among the three loss functions that we have defined and evaluated, the CDFL—specified to maximize the discrimination margin between valid and high-scoring invalid paths—gives the best performance. A comparison with the state of the art on both action segmentation and action alignment tasks, for the Breakfast Action, Hollywood Extended and 50Salads datasets, supports our novelty claim that using our CDFL in training gives superior results than a loss function estimated on a single inferred segmentation, as done by prior work. Our results on both action segmentation and action alignment tasks also demonstrate advantages of considering many candidate segmentations in neighbor-windows around the initial video segmentation, and maximizing the margin between all valid and hard invalid segmentations. Our small increase in complexity relative to that of related work seems justified considering our significant performance improvements.

**Acknowledgement.** This work was supported in part by DARPA XAI Award N66001-17-2-4029 and AFRL STTR AF18B-T002.
References


