An ordering loss approach to weakly supervised activity detection

Atal Narayan Sahu
KAUST
Thuwal, Saudi Arabia
atal.sahu@kaust.edu.sa

Abstract

We focus on the task of weakly supervised action segmentation in a video with the supervision as the sequence of actions happening in the video. We follow a novel approach in which we decompose the action sequence in a video into ordered action pairs. We then propose to use ordering based loss functions for each pair. The loss functions enforce that for each pair, the classifier's prediction respects the pair ordering in given action sequence. In our experiments, we find that our proposed approach does not improve upon the existing approaches.

1. Introduction

Activity detection is the problem of localizing and classifying actions in long untrimmed videos. A fully supervised case has each each frame in the training dataset annotated with its ground truth label. However, obtaining such annotations is costly and tedious, since a video typically consists of hundreds of frames. In order to circumvent this issue, [6, 1] proposed training classifiers with weaker supervision.

The literature primarily consists of two types of weak supervision: action set [6], and action sequence [1] based weak supervision. In action set based weak-supervision, the classifier is only provided the set of actions in a given training video for supervision. In action sequence based weak supervision, which is also referred to as transcript based weak supervision, the classifier is provided with the sequence of actions occurring in the video as supervision. If an action occurs multiple times in a video, then the transcript consists of both the occurrences of that action. Clearly, transcripts are a stronger form of supervision than action sets.

Our work focuses on transcript based supervision for activity detection. Existing works for this problem have mainly followed two routes: 1) **Pseudo ground-truth based approaches** [3, 8] iterate over constructing pseudo ground truth for training the classifier, and subsequently refining the pseudo ground truth from the classifier's predictions. The refined pseudo ground truth is used in the next iteration, and the whole process is repeated till some stopping criterion is met. 2) **Transcript loss based approaches** [2, 5] propose transcript based discriminative losses, which discriminate between the *valid paths*: the paths which follow the ground-truth transcript against the *invalid paths*: the paths which don't follow the transcript. Both of the approaches have their drawbacks. An issue with the pseudo-ground truth approach is that the training is unstable, as the model is as good as the pseudo ground truth is, and one cannot determine the quality of the pseudo ground truth. Similarly, while the discriminative loss based strategies achieve state-of-the art, they involve a costly sampling of invalid paths during training.

In contrast, we follow a novel ordering loss approach by decomposing the transcript into ordered action pairs, and then use ordering losses which ensure that each pair respects its order. In order to follow our ordering loss approach, we devise and experiment with a variety of ordering losses, and report our results.

2. Related work

There has been an excellent progress on transcript based weakly supervised action detection over the past few years. Since we don't have access to the ground-truth segmentation during training, prior works are mainly based on generating pseudo ground-truth segmentation or incorporate a transcript based loss. We provide a brief review for each type below:

- **Pseudo Ground-truth generation:** [3] start from a uniform frame distribution to each action in the ground-truth and use an iterative soft boundary assignment (ISBA) mechanism to periodically refine the pseudo ground-truth as training progresses. Similarly, [7] and [8] use Viterbi decoding with the help of a length-model and action priors to generate a pseudo ground-truth at each iteration. They then use the cross-entropy loss with this pseudo ground-truth to train their model.

- **Transcript based loss**: [2] and [5] incorporate discriminative transcript based loss which favors the valid paths: *the paths which follow the ground-truth transcript* against the invalid paths: *the paths which don't follow the ground truth transcript*. Evaluating these transcript losses also involves a costly viterbi decoding step and sampling of invalid and valid paths.

- **MuCon** [9] jointly train two predictors both performing action segmentation using a novel Mutual Consistency (MuCon) loss. The predictors are a frame-wise classifier and an RNN which predicts the transcript and lengths of each action in the transcript. Training the RNN for transcript prediction can be done using the ground-truth transcripts, while the action-length prediction task of RNN is coupled with the frame-wise classifier using the MuCon loss. MuCon enforces the predictions from the frame-wise classifier and RNN to be consistent with each other.

Our approach neither uses pseudo ground truth based training, nor do we construct a discriminative transcript based loss. We also don't couple multiple classifiers as in MuCon, although we believe our ordering losses should compose well with MuCon's approach.

3. Notations

Superscripts refer to frame index in a video and subscripts refer to action index in a transcript. We use $P(\cdot)$ to denote probabilities.

4. Ordering loss functions for action pairs

We approach the problem from an altogether new perspective. Consider a transcript with the action sequence as $(a_1, a_2, \dots, a_n)$. Assume for now that an action occurs in the transcript only once. Then, one can extract ordered action pairs (a_i, a_j) from the transcript, such that action i occurs before action j in the transcript. There would be a total of $\binom{n}{2}$ such pairs which if given can completely recover the transcript. We want to construct an ordering based loss to ensure that our classifier's prediction respects each pair's order. Once we construct an ordering loss for a pair, the loss for a whole video can just be the average loss across ordered pairs in the video.

A serious drawback of our approach is in the case when an action occurs more than once in the video. For this action, we can only extract ordered pairs if the other action has all of its occurrences either before or after this action. But following this route has advantages as well. Firstly, the training would not be unstable as in the case of pseudo-ground truth based approaches. Secondly, this approach does not require a search over the valid and invalid paths or viterbi decoding, and hence is significantly faster during training as we would see in our experiments. Lastly, finding an ordering loss is a more general problem with use cases such as for learning deep-embeddings.[10]

In order to construct an ordering loss, we derive temporal probabilities of occurrence for each action in the transcript. Let a video have its frames as $f^1, f^2, \dots f^T$, for a total of T frames. Then by introducing temporal probabilities, we intend to find the likely video regions where the classifier predicts an action to be present. Let for an action a_i, p_i^t denote its temporal probability, and ϕ_i^t its cumulative distribution function at frame $t \in [T]$. Similarly, let X_i denote a random variable following probabilities p_i^t. In the next section, we construct loss functions for an ordered pair (a_l, a_r), where a_l denotes the left action, and a_r denotes the right action. Note that all the losses that we construct are minimization losses.

4.1. Mean Distance loss

A straightforward approach is to make the mean occurrence of action a_l as much to the left of the mean occurrence of action a_r as possible, i.e.

$$\mathcal{L}_{MDL}(a_l, a_r) = \mathbb{E}\left[X_l\right] - \mathbb{E}\left[X_r\right] \tag{1}$$

This is a simple loss, and $\mathcal{L}_{MDL}(a_l, a_r)$ would be negative if the classifier's temporal predictions follows the action order. But it has some major drawbacks. First, the loss is unstable. Even when $P(X_l >= X_r) = 0$, i.e. when the classifier predicts zero probability of action a_l occuring after action a_r, the loss would have non-zero gradients. Second, when the loss is summed for all the ordered pairs for a video, $\mathbb{E}\left[X_l\right]$ for many actions would cancel out. Consider a transcript with three distinct actions (a_1, a_2, a_3), then the loss would be

$$\begin{aligned}
& \mathcal{L}_{MDL}(a_1, a_2, a_3) \\
= \; & \frac{1}{\binom{3}{2}}(\mathcal{L}_{MDL}(a_1, a_2) + \mathcal{L}_{MDL}(a_2, a_3) + \mathcal{L}_{MDL}(a_1, a_3)) \\
= \; & \frac{1}{3}(\mathbb{E}\left[X_1\right] - \mathbb{E}\left[X_2\right] + \mathbb{E}\left[X_2\right] - \mathbb{E}\left[X_3\right] + \mathbb{E}\left[X_1\right] - \mathbb{E}\left[X_3\right]) \\
= \; & \frac{1}{3}(2\mathbb{E}\left[X_1\right] - 2\mathbb{E}\left[X_3\right]).
\end{aligned}$$

Hence, the classifier does not get any feedback about the location of a_2 as $\mathbb{E}\left[X_2\right]$ does not occur in the final loss calculation.

4.2. Histogram loss

We now focus on removing the drawbacks of $\mathcal{L}_{MDL}$. The problem of enforcing "order" between two distributions has also been considered to learn deep embeddings by Ustinova et al. [10]. Their setting involves learning semantically meaningful embeddings for images. Put simply, they want to ensure that embeddings of different images of the same person are similar, and that of different persons are dissimilar. They compute similarity between pairs of images, where positive pairs correspond to images of the same person, and negative pairs to distinct persons. Using a similarity measure, they compute the histograms(or distributions) of similarities of positive and negative pairs. Let the probability mass function of the positive and negative pairs be p^+ and p^- respectively. Similarly, let the cumulative distribution functions be ϕ^+ and ϕ^- respectively. Let similarity values lie in $[-1, 1]$. Consider the following probability:

$$p_{reverse} = \int_{-1}^{1} p^-(x)\phi^+(x)dx = \mathbb{E}_{x\sim p^-}\left[\phi^+(x)\right] \tag{2}$$

In simple terms, $p_{reverse}$ is the probability of the similarity in a random negative pair to be more than the similarity in a random positive pair (hence referred to as *reverse*). We use a similar construct for our problem:

$$\mathcal{L}_{hist}(a_l, a_r) = \mathbb{E}_{t\sim p_l}\left[\phi_r^t\right] = \sum_{t=1}^{T} p_l^t \phi_r^t \tag{3}$$

Clearly $\mathcal{L}_{hist}$ does not have the second drawback of $\mathcal{L}_{MDL}$, since the terms won't cancel out when averaging over pairs. Hence every pair in the transcript would contribute to the loss.

4.3. Difference random variable derived loss functions

In this section, we derive the difference random variable $X_l - X_r$ assuming X_l and X_r are independent. Using the difference rv, we first construct the difference probability loss: $\mathcal{L}_{DPL}(a_l, a_r) = P(X_l - X_r \geq 0)$. We then show that the probability loss $\mathcal{L}_{DPL}$ is an alternative way to compute the histogram loss. We also show how to derive richer loss functions from the difference random variable.

Let us construct the difference random variable $X_l - X_r$. We have, for $\Delta \in \{0, \ldots, T-1\}$,

$$\begin{aligned} & P(X_l - X_r = \Delta) \\ = & \sum_{t=1}^{T-\Delta} P((X_l, X_r) = (t+\Delta, t)) \\ = & \sum_{t=1}^{T-\Delta} P(X_l = t+\Delta) \cdot P(X_r = t) \quad \text{(independence)} \\ = & \sum_{t=1}^{T-\Delta} p_l^{t+\Delta} p_r^t \end{aligned}$$

Similarly, $P(X_l - X_r = -\Delta) = \sum_{t=1}^{T-\Delta} p_l^t p_r^{t+\Delta}$. Then, the difference probability loss is

$$\mathcal{L}_{DPL}(a_l, a_r) = P(X_l - X_r \geq 0) = \sum_{\Delta=0}^{T-1} P(X_l - X_r = \Delta)$$

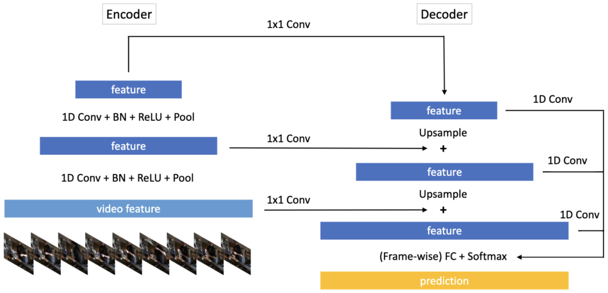

Figure 1. The TCFPN model (figure from [3])

Theorem 1. *For an ordered action pair* (a_l, a_r), $\mathcal{L}_{DPL}(a_l, a_r)$ *is equivalent to* $\mathcal{L}_{hist}(a_l, a_r)$.

However, computing the difference variable allows for using richer loss functions with different weights to different Δs. For e.g., consider the following reweighted versions of $\mathcal{L}_{DPL}$.

$$\mathcal{L}_{WDPL}(a_l, a_r) = \sum_{\Delta=0}^{T-1} w(\Delta) \cdot P(X_l - X_r = \Delta) \tag{4}$$

Intuitively, higher differences should be penalized more because they account for higher errors in temporal predictions. However, we remind ourselves that we have assumed independence of the two actions' probabilities, which is not actually true. In our experiments, we investigate both increasing and decreasing weight functions and report our findings.

5. Empirical study

5.1. Model

We use the open-source Temporal Convolution Feature Pyramid Network (TCFPN) [1] model for our experiments. While there has been significant progress in the recent years on the problem of transcript based weakly supervised action segmentation, our choice of the above repository is governed by one of the following reasons: a) The newer improvements are not architectural, but based on new innovative training strategies. These improvements use rather shallow networks, such as a single hidden layer RNN [5]. b) Some newer newer improvements don't have their code available online.

The TCFPN model is depicted in figure 1. It is a Temporal Convolutional Network with an encoder-decoder architecture. The depth of the encoder and decoder layers is three in our experiments. The model outputs frame-wise classification probabilities. We refer the readers to the original paper for further information on the architecture of the model.

[1] https://github.com/Zephyr-D/TCFPN-ISBA

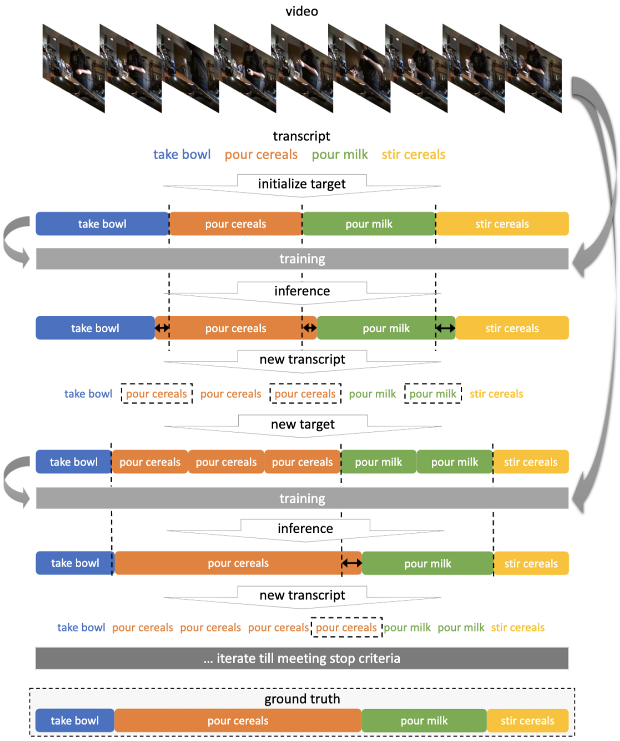

Figure 2. The ISBA mechanism (figure from [3])

In order to train TCFPN in a weakly supervised setting, Ding et al. [3] take a pseudo ground truth based approach. They introduce a mechanism called Iterative Soft Boundary Assignment (ISBA), depicted in figure 2. They start from an initial pseudo ground truth, which is constructed by uniformly distributing the transcript's actions across a video. Then ISBA iterates over training the classifier using the updated pseudo ground truth, and updating the pseudo ground truth using the trained classifier's predictions. They use performance of the classifier on predicting the action set as the stopping criterion. That is, if the action set prediction performance stagnates or starts to degrade, then they stop training.

For our ordering based loss functions, we need to extract temporal probabilities for actions occurring in the ground truth sequence from the classifier. Our procedure for the same is depicted in figure 3. For an action a_i in the transcript, we normalize the frame-wise probabilities for action a_i in the time (frame) domain. Let $p_\theta(i,t)$ denote the prediction probability output by the classifier for action a_i. Then, we experiment with the following normalization strategies:

- **Linear normalization:** $p_i^t = \dfrac{p_\theta(i,t)}{\sum_{t=1}^{T} p_\theta(i,t)}$,

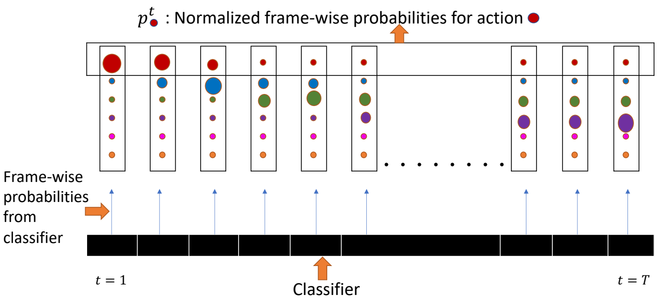
$p^t_{\bullet}$: Normalized frame-wise probabilities for action ●
Frame-wise probabilities from classifier
$t = 1$
Classifier
$t = T$

Figure 3. Extracting temporal probabilities for an action from the classifier.

- **Softmax:** $p_i^t = \dfrac{exp(p_\theta(i,t))}{\sum_{t=1}^{T} exp(p_\theta(i,t))}$.

5.2. Dataset and metrics

We test on the Breakfast dataset [4] which is the standard dataset for the problem of transcript based weakly supervised action segmentation. The breakfast dataset comprises of 1712 videos corresponding to 10 breakfast dishes, such as salad, pancake, coffee, etc. There are a total of 48 actions in the dataset, hence a random predictor on the dataset would correctly predict a frame with probability 2.08%. The dishes are made by a total of 52 individuals, and their is some variability in the recipe used by different individuals even of the same dish.

Similar to [3], our experiments consist of the following metrics:

- **Frame-wise accuracy (Acc.):** Ratio of testing frames correctly predicted.
- **Frame-wise accuracy without background (Acc.-b.g.):** Ratio of testing frames correctly predicted by not considering the *background* frames.
- **Intersection over Union (IoU):** Given a ground-truth action interval I^*, and a classifier's prediction interval I, $|I \cap I^*|/|I \cup I^*|$.
- **Intersection over Detection (IoD):** Under the same notation as above, IoD denotes $|I \cap I^*|/|I|$.

5.3. Complimentary loss functions

As an action sequence corresponding to a video also provides us information about the action set occurring in the video, we can simply use action set based loss functions in conjunction with our ordering based loss functions. Below we present and use two such loss functions from Paul et al [6].

5.3.1 MILL: Multiple Instance Learning Loss

The k-max multiple instance learning loss [6] is based on the intuition that if an action is present in a video's action set, then it should occur in atleast k frames in the video, where k is a hyper parameter. A critical issue is that we cannot determine the corresponding frames for each action just from the action set. Paul et al. [6] use the classifier's predictions as an indication of an action's occurrence. For an action a_i and the corresponding classifier's probabilities $\{p_i^t\}_{t=1}^T$, we select the Top-k frames with highest action a_i probabilities, and minimize the negative log likelihood of the average of these probabilities. That is, for a given action a_i in ground truth transcript, let $p_i^{MILL} = \frac{\sum_{i \in Top-k(\{p_i^t\}_{t=1}^T)} p_i^t}{k}$, we have

$$\mathcal{L}_{MILL}(a_i) = -log(p_i^{MILL}) \tag{5}$$

For a given video, the loss is averaged over all actions in the action set. The parameter k varies for each video, and is determined as a fraction of the total frames T for a video. Choosing $s > 1$, we have $k = \max(\lfloor \frac{T}{s} \rfloor, 1)$. In our experiments, we fix $s = 20$ as it provides as with the best results.

5.3.2 CASL: Co-Activity Similarity loss

Till now, all the losses that we discussed corresponded to a single video. However, one can also think of inter-video loss functions. The Co-Activity similarity loss [6] is constructed with the following intuition: If two videos m and n have the same action a_j in their action set, then the similarity between features corresponding to regions where action a_j occurs in videos m and n should be higher than the similarity of action a_j region features in video m and non-action a_j region features in video n (and vice-versa).

Again, one does not know the regions where action a_j occurs just from the action set. We would use the classifier to provide us soft information via temporal probabilities about action $a_j^{\prime}s$ presence or absence in a frame in the video. Using the decoder features from the layer before *FC+Softmax*, we now compute high and low region features for action a_j in video m (${}^Hf_m^j$ and ${}^Lf_m^j$ respectively). Let $\{d_m^t\}_{t=1}^T$ denote the extracted features of video m corresponding to action a_j. We hereby remove the subscript m for notational convenience. We have the high region features as:

$${}^Hf^j = \frac{p_j^t d^t}{\sum_{t=1}^T p_j^t} \tag{6}$$

Similarly, we have the low region features as:

$${}^Lf^j = \frac{(1-p_j^t)d^t}{T - \sum_{t=1}^T p_j^t}, \tag{7}$$

where we use the softmax normalization to extract the temporal probabilities p_j^t. Using the constructed features, we define the Co-Activity Similarity Loss for videos m and n sharing action a_j as:

$$\begin{aligned}\mathcal{L}_{CASL} = \quad & \frac{1}{2}\{\max(0, d[{}^Hf_m^j, {}^Hf_n^j] - d[{}^Hf_m^j, {}^Lf_n^j]) \\ & + \max(0, d[{}^Hf_m^j, {}^Hf_n^j] - d[{}^Lf_m^j, {}^Hf_n^j])\}\end{aligned}$$

Method	Acc.	Acc.-bg	IoU	IoD
$\mathcal{L}_{MDL}$	15.88	15.93	12.31	32.11
$\mathcal{L}_{hist}$	23.36	20.71	16.07	38.29
$\mathcal{L}_{WDPL}$ dec	20.36	15.34	12.04	30.48
$\mathcal{L}_{WDPL}$ inc	21.57	17.56	12.61	32.77
Pseudo GT	**37.1**	**36.9**	**23.1**	**40.6**

Table 1. Linear normalization experiments ($\mathcal{L}_{MILL}$ alone achieves an accuracy of 13.1%). TCFPN's original Pseudo-ground truth based ISBA mechanism performs the best.

Method	Acc.	Acc.-bg	IoU	IoD
$\mathcal{L}_{hist}$	23.36	20.71	16.07	38.29
$\mathcal{L}_{hist} + \mathcal{L}_{CASL}$	25.01	22.13	17.9	39.56

Table 2. Adding $\mathcal{L}_{CASL}$ improves performance for $\mathcal{L}_{hist}$ when using linear normalization for temporal probabilities.

5.4. Results

Let an ordering loss be denoted by $\mathcal{L}_{ord}$. Then, the final loss function is given as follows:

$$\mathcal{L}_{final} = \lambda_{ord} \cdot \mathcal{L}_{ord} + \lambda_{MILL} \cdot \mathcal{L}_{MILL} + \lambda_{CASL} \cdot \mathcal{L}_{CASL} \tag{8}$$

We tune the loss coefficients using grid-search in our experiments.

5.4.1 Linear Normalization

In table 1 we present the results with linear normalization for extracting temporal probabilities. In these experiments, we don't add $\mathcal{L}_{CASL}$, and tune the coefficients λ_{MILL} and λ_{ord} for each loss. We find that the $\mathcal{L}_{MDL}$ performs the worst and achieves an average accuracy of 15.88%. $\mathcal{L}_{hist}$ performs better and achieves an accuracy of 23.36%. Sadly, introducing either linearly increasing ($\mathcal{L}_{WDPL}$ inc) or decreasing weights ($\mathcal{L}_{WDPL}$ dec) in the difference probability loss degrades the performance, and we suspect this is due to the violation of the independence assumption in constructing the difference random variable. In summary, our best performing ordering loss is $\mathcal{L}_{hist}$ (histogram aka difference probability loss), but we are still far from the original pseudo-ground truth based ISBA mechanism's performance.

In table 2, we couple our best performing histogram loss from the previous table with the co-activity similarity

Method	**Acc.**	**Acc.-bg**	**IoU**	**IoD**
$\mathcal{L}_{MDL}$	13.38	13.93	9.524	18.16
$\mathcal{L}_{MDL}$+$\mathcal{L}_{MILL}$	30.5	28.12	19.31	35.1
$\mathcal{L}_{hist}$ + $\mathcal{L}_{MILL}$	17.61	16.51	11.19	20.81
Pseudo GT	**37.1**	**36.9**	**23.1**	**40.6**

Table 3. Softmax normalization experiments. $\mathcal{L}_{MDL} + \mathcal{L}_{MILL}$ has good performance.

loss. We find that adding $\mathcal{L}_{CASL}$ further improves average framewise accuracy by 1.65%, achieving an accuracy of 25.01%.

5.5. Softmax Normalization

We now present our results with softmax normalization for extracting temporal probabilities. Performing softmax normalization instead of linear in the temporal dimension would make the temporal probabilities closer to a uniform distribution. This can be explained using the following simple example. Consider only two frames: $t \in \{1, 2\}$ with classifier's predicted probabilities for action a_i as $p_\theta(i, 1) = 0.99$, and $p_\theta(i, 2) = 0.02$. If we perform linear normalization, then the temporal probabilities turn out to be $p_i^1 = 0.980$, and $p_i^2 = 0.019$ when ignoring the smaller decimals. If we perform softmax normalization, then the temporal probabilities become $p_i^1 = 0.725$, and $p_i^2 = 0.274$ which is more closer to a uniform distribution. Essentially the exponential function has the range $[1, e]$ when the inputs are values in the range $[0, 1]$ (probabilities in our case), which makes the softmax closer to a uniform distribution.

In softmax normalization experiments, we perform one more softmax for the input probabilities to $\mathcal{L}_{MILL}$. The average accuracy without double softmax for all ordering losses turn out to be in $12.01\% - 14.03\%$, which is also the performance of $\mathcal{L}_{MILL}$ alone. We suspect that this is due to $\mathcal{L}_{MILL}$ quickly converging to a solution, and the ordering losses doing some local modifications around $\mathcal{L}_{MILL}$. Having another softmax for $\mathcal{L}_{MILL}$ will also make its input more closer to a uniform distribution, and thus we hope that the loss surfaces have similar curvature. Furthermore, $\mathcal{L}_{CASL}$ with/without similar hacks as $\mathcal{L}_{MILL}$ didn't improve performance for any ordering loss in this case, and hence we are not reporting experiments with $\mathcal{L}_{CASL}$ here.

In table 3, we report the results using softmax normalization. $\mathcal{L}_{hist}$ in conjunction with $\mathcal{L}_{MILL}$ has an accuracy of 17.61%. Surprisingly, $\mathcal{L}_{MDL}$ has an accuracy of only 13.38% when used alone, but gives an accuracy of 30.5% when used in conjunction with $\mathcal{L}_{MILL}$.

5.5.1 Ordering loss in conjunction with TCFPN pseudo-ground truth training

Another question that needs to be addressed is if our ordering loss functions can improve or accelerate the pseudo-ground truth based approach's performance. In table 4, we find that the performance of pseudo ground truth approach degrades when used in conjunction with $\mathcal{L}_{MDL}$ (softmax normalization).

Method	**Acc.**	**Acc.-bg**	**IoU**	**IoD**
Pseudo GT + $\mathcal{L}_{MDL}$	32.5	29.7	18.66	34.89
Pseudo GT	**37.1**	**36.9**	**23.1**	**40.6**

Table 4. Adding $\mathcal{L}_{MDL}$ with the pseudo ground truth approach degrades performance.

5.5.2 Training time

In table 5, we report the training times for one split of the breakfast dataset for our ordering loss functions using one NVIDIA V100 GPU. In calculating these training times, we discard the data-loading time, which is around 3 minutes for the breakfast dataset. We find significantly faster convergence for our ordering loss functions in comparison to the pseudo-ground truth based approach. However, these results are for time to convergence, and ISBA converges to a significantly better model. In 14.7 minutes, ISBA achieves an accuracy of 30%, and hence is better to use than $\mathcal{L}_{hist}$. The true improvement is for $\mathcal{L}_{MDL}$ with softmax normalization, which achieves similar performance in only 4 minutes.

5.5.3 Concluding remarks on experiments

Our experiments till now have shown that ordering losses didn't improve upon the pseudo ground truth approach. However, there are a few modifications that we would try in future. Intuitively, ordering losses such as $\mathcal{L}_{hist}$ should work even if the input is not a probability distribution, but has positive values in a bounded range. We plan to add temporal attention mechanism and provide attention maps as input to ordering loss instead of the extracted temporal probabilities. The primary reason to do so is because we want to decouple the temporal distributions of actions, which clearly does not happen in our present setting.

Finally, although there are papers which have further improved performance on transcript-based action detection[5, 9], our focus for this project was on following an ordering loss approach. For instance, [5] achieves state of the art with an average accuracy of 50.2% on breakfast dataset by training an RNN using a discriminative loss. But their total training time is around 28 hours[9] on one V100 GPU since their procedure involves costly viterbi-decoding and a sampling step. We chose TCFPN for our experiments since it was a fast temporal convolution network with a publicly available codebase.

Method	**Training time (mins)**
Pseudo GT	133
$\mathcal{L}_{hist}$ (linear)	14.7 (9 x)
$\mathcal{L}_{MDL}$ (softmax)	**4 (33.25 x)**

Table 5. Training time improvements with respect to the baseline pseudo ground truth based ISBA mechanism.

6. Conclusion

We followed an ordering loss approach for weakly supervised action segmentation. We devised various ordering losses, and drew connections with the existing ones. In our experiments, we found our approach to under perform with respect to the existing pseudo ground truth approach. However, ordering losses have use cases beyond our setup, and an interesting direction of future research is to use the richer weighted difference probability loss for learning deep embeddings.

References

[1] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *European Conference on Computer Vision*, pages 628–643. Springer, 2014. 1

[2] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[3] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 3, 4

[4] H. Kuehne, A. B. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR)*, 2014. 4

[5] Jun Li, Peng Lei, and Sinisa Todorovic. Weakly supervised energy-based learning for action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 3, 6

[6] Sujoy Paul, Sourya Roy, and Amit K. Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 1, 4, 5

[7] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[8] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[9] Yaser Souri, Mohsen Fayyaz, and Juergen Gall. Weakly supervised action segmentation using mutual consistency. *CoRR*, abs/1904.03116, 2019. 2, 6

[10] Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 4170–4178. Curran Associates, Inc., 2016. 2

A. Proofs

A.1. Proof of Theorem 1 (equivalence of $\mathcal{L}_{hist}$ and $\mathcal{L}_{DPL}$.)

We have,

$$\begin{aligned}
\mathcal{L}_{hist}(a_l, a_r) &= \sum_{t=1}^{T} p_l^t \phi_r^t \\
&= \sum_{t=1}^{T} p_l^t \sum_{t'=1}^{t} p_r^{t'} \\
&= \sum_{t'=1}^{T} \sum_{t=t'}^{T} p_l^t p_r^{t'} \\
&= \sum_{t'=1}^{T} \sum_{\Delta=0}^{t'-\Delta} p_l^{t'+\Delta} p_r^{t'} \\
&= \sum_{\Delta=0}^{T-1} \sum_{t'=1}^{T-\Delta} p_l^{t'+\Delta} p_r^{t'} \\
&= \sum_{\Delta=0}^{T-1} P(X_l - X_r = \Delta) \\
&= P(X_l - X_r \geq 0) \\
&= \mathcal{L}_{DPL}(a_l, a_r)
\end{aligned}$$