Weakly-Supervised Dense Action Anticipation

Haotong Zhang[1]
haotongz@u.nus.edu
Fuhai Chen[2]
cfh3c@nus.edu.sg
Angela Yao[2]
ayao@comp.nus.edu.sg

[1] Department of Mathematics
National University of Singapore
Singapore

[2] Department of Computer Science
National University of Singapore
Singapore

Abstract

Dense anticipation aims to forecast future actions and their durations for long horizons. Existing approaches rely on fully-labelled data, *i.e*. sequences labelled with *all* future actions and their durations. We present a (semi-) weakly supervised method using only a small number of fully-labelled sequences and predominantly sequences in which only the (one) upcoming action is labelled. To this end, we propose a framework that generates pseudo-labels for future actions and their durations and adaptively refines them through a refinement module. Given only the upcoming action label as input, these pseudo-labels guide action/duration prediction for the future. We further design an attention mechanism to predict context-aware durations. Experiments on the Breakfast and 50Salads benchmarks verify our method's effectiveness; we are competitive even when compared to fully supervised state-of-the-art models. We will make our code available at: https://github.com/zhanghaotong1/WSLVideoDenseAnticipation.

1 Introduction

Anticipating human actions is critical for real-world applications in autonomous driving, video surveillance, human-computer interaction, *etc*. According to the prediction horizons, the anticipation task is mainly investigated in two tracks: next-action anticipation [6, 8, 14, 28, 31, 34, 43] and dense anticipation [14, 28, 43]. Next action anticipation predicts upcoming actions τ seconds in advance, where the value of τ is considered as 1 in many recent works. Dense anticipation predicts multiple actions into the future and their durations for long horizons of up to several minutes or an entire video.

Our paper focuses on the more challenging dense anticipation task where all existing methods [14, 28, 43] are fully supervised. Annotating videos for the fully supervised version of this task can be tedious, as it requires labelling the full set of actions in the subsequent sequence as well as their start and end times. In real-world videos, sequences are more likely to be labelled or tagged only at specific events. These tags are incomplete and instantaneous, *i.e*. not present at every action and without duration information. This motivates us to develop a weakly supervised dense anticipation framework that learns from video sequences with an

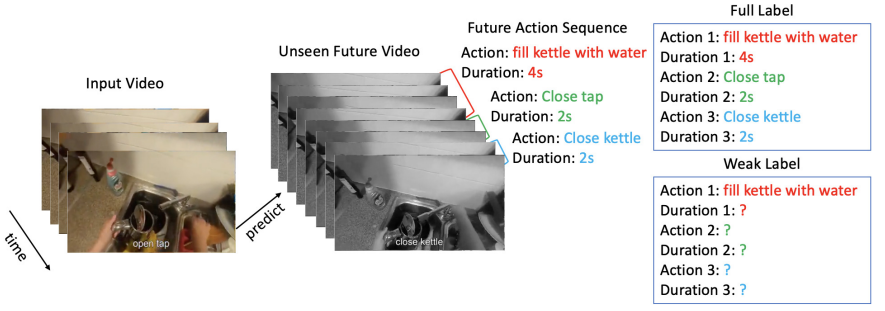

Figure 1: Dense anticipation with full supervision vs. weak supervision. The fully supervised label contains all the actions in the future video sequence as well as their durations. In this work, we consider a weak label in which only the first action label without any duration information is available. Our proposed framework is both semi- and weakly-supervised. We use a small set of fully-labelled videos, while the remainings are weakly-labelled.

incomplete set of action and duration labels. Specifically, we aim to learn from a small set of fully-labelled data and predominantly from weak labels in which the video segment is annotated only with the first action class of the anticipated sequence (see Fig. 1). This can greatly reduce the labelling effort as now we only need to provide the class label of a single action instead of all frames in the sequence.

In practice, this type of weak label is akin to the *time-stamp annotations* used in weakly-supervised temporal action segmentation, in which an arbitrary frame from each action segment is labelled [9, 12, 49]. When annotating timestamps, annotators quickly go through a video and press a button when an action is occurring. This is ∼6x faster than marking the exact start and end frames of action segments [12] and still provides strong cues to learn effective models for action segmentation.

In our case, our weak label can be viewed as an incomplete version of the full label since it has only one (the first) of the full set of action labels, and no duration labels. Since each action label and action duration are all treated as separate terms in the loss for conventional anticipation methods [14, 34, 43], a naive route to learn would be to ignore any missing labels from the loss. This option, while simple, does not fully leverage the data of the weakly-labelled set. We opt instead to learn an auxiliary model to generate pseudo-labels for the missing labels. The use of pseudo-labelling has become popular in unsupervised and semi-supervised learning [19, 29, 38, 45] and has been successful for tasks like image classification [6, 18, 36, 48] and segmentation [23, 41, 47]. Inspired by these works, we propose a framework for learning a primary and conditional module for (semi-) weakly-supervised dense action anticipation. The conditional module is learned on a small fully-labelled training set to generate pseudo-labels for a larger weakly-labelled training set. The pseudo-labelled weak data is then applied to learn the primary anticipation module which will be used during inference.

Directly learning on the outputs of an auxiliary model is often not better than learning on the limited set of provided labels as it does not add new knowledge into the system. The phenomenon is referred to as confirmation bias [40]; extending previous solutions such as label smoothing [20] or label sampling and augmentation [2, 7, 42] is non-trivial for sequence data. As such, we introduce an adaptive refinement method which learns refined sequence labels based on the predictions of the primary and conditional module. In our

experimentation, we have observed that the accuracy of dense anticipation is highly sensitive to having the correct duration prediction, especially in the earlier anticipated actions[1]. We are therefore motivated to ensure that the anticipated durations are correct. To that end, we introduce an additional duration attention module applicable to recursive dense anticipation methods [14, 43]. We compute an attention score between the observed video context and the hidden representation at each prediction step to explicitly emphasize the correlations, which greatly improves the duration accuracy.

The contributions of this paper are summarized as follows:

1. We explore a novel and practical weakly-supervised dense anticipation task and propose an adaptive refinement method to make the most of weakly-labelled videos while using only a small number of fully-labelled videos.

2. We propose an attention scheme for predicting the duration of the anticipated actions which better accounts for the action correlations.

3. Our semi-supervised framework is flexible and applicable to a variety of dense anticipation backbones. The duration attention scheme serves as a plug-and-play module to improve the performance of recursive anticipation methods. Evaluation on standard benchmarks shows that our weakly supervised learning scheme can compete with state-of-the-art fully supervised approaches.

2 Related Work

Action recognition is the hallmark task of video understanding. In standard action recognition settings, short, trimmed video clips are classified with action labels. In contrast, action anticipation is applied to longer, untrimmed video sequences and aims to predict future actions *before* they occur. The task in next action anticipation is to predict the upcoming action τ before it occurs. Various architectures ranging from recurrent neural networks (RNNs) [1, 4, 13, 35], convolutional networks combined with RNNs [34], to transformers [37] are proposed. The main focus of these works is to extract relevant information from the observations to predict the label of the action starting in τ seconds, varying between zero [33] to 10s of seconds [15]. Other models leverage external cues such as hand movements to help with the anticipation task [11, 27].

Dense action anticipation predicts *all* subsequent actions and their durations for longer horizons of the unobserved sequence. Recursive methods [14, 43] use an encoder to extract visual features from the observed sequence and use an RNN as a decoder to predict future actions and their duration sequentially. As recursive predictions may accumulate and propagate errors, Ke *et al*. [28] anticipates actions directly for specific future times in a single shot. When it comes to duration anticipation, all previous methods are relatively simple in that they apply a linear layer on top of the features of observed or predicted actions. Only past action features are used, without taking action correlations into account. Intuitively, actions with higher correlations with current action tend to influence more on current action's duration. Consequently, our method improves on previous works by introducing an attention mechanism for duration anticipation.

To date, all methods for dense anticipation [14, 28, 43] follow a fully supervised setting and require extensive annotations for learning. Driven by the laborious demand of fully labelled data in computer vision, some researchers focus on weakly- or semi-supervised

[1]Consider a ground truth sequence of AABBCCDD where each letter is the action of a frame; a prediction of AAAABBCCDD would score a mean-over-classes of only 0.25 since all B, C and D frames are misaligned.

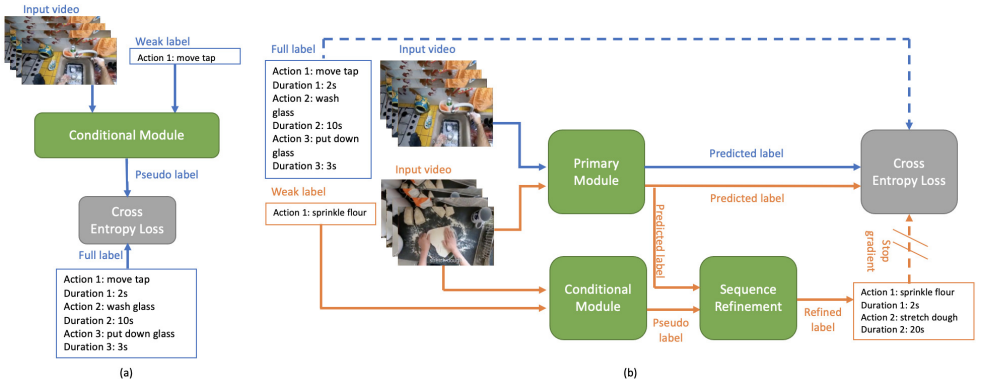

Figure 2: Method overview. (a) The conditional module is trained on the small set of fully-labelled data to generate pseudo-labels. Once trained, it remains fixed, and does not contribute gradients in the following steps. (b) The primary module is trained on the small set of fully-labelled data and the large set of weakly-labelled data with the first future action label as the incomplete label. The weak labels are augmented into full pseudo-labels by refining the outputs of the conditional module.

learning to reduce annotation workload [21, 24, 26, 44]. Previously, [46] apply a weakly-supervised model on forecasting future action sequences, where only action sequences rather than frame-wise labels are provided as coarse labels. They combine the attention scheme with GRU to recurrently predict action labels with more focus on related observed actions, which is similar to our duration attention. Our work is similar in spirit to the teacher-student model [3, 30] which also uses an auxiliary model to support training. However, we do not explicitly enforce label consistency between the two models and instead use a third refinement module to directly improve the pseudo-labels. Pseudo-labels are widely used in weak supervision [25, 39, 40]. Most often they are only propagated for unlabelled or semi-labelled data. We also generate for fully-labelled data and by minimizing the distance between ground truth and improved pseudo labels by our refinement module, we make the model adaptive refine the accuracy of pseudo labels.

3 Method

Our proposed framework is both semi- and weakly-supervised. It is trained on a small set of fully labelled videos and a large set of weakly labelled videos with only the first action in the anticipated sequence. The model is comprised of three components: a primary module used during inference (Sec. 3.3), a conditional module for generating pseudo-labels (Sec. 3.2), and a sequence refinement module (Sec. 3.4) to refine the estimated pseudo-labels.

We treat the primary and conditional modules as black-box encoder-decoders, where the observed video is encoded into features, while the decoder generates the anticipated action labels and durations. As the proposed framework is general, we can use any previously proposed dense anticipation model [14, 28, 43] as a backbone. The training procedure can be broken down into two stages. The conditional module is trained initially on the fully-labelled set $\mathcal{F}$ so that it can be used to generate pseudo-labels for the weakly-labelled data. The combined set of the fully-labelled and the pseudo-labelled weak data $\mathcal{W}$ then merged to

train the primary module. Directly using the pseudo-labels may result in confirmation bias, as these labels are generated from a model which is learned only on the small set of fully-labelled data. Therefore we refine the pseudo-labels with a sequence refinement module which is learned simultaneously with the primary module. Fig. 2 illustrates an overview.

3.1 Preliminaries

For a given video, $\mathbf{x} = \{x_1, \ldots, x_t, \ldots, x_T\}$ denotes the set of T observed frames. Dense anticipation aims to predict the future M action labels $\mathbf{c} = \{c_1, \ldots, c_m, \ldots, c_M\}$ and associated durations $\mathbf{d} = \{d_1, \ldots, d_m, \ldots, d_M\}$ for frames $T+1$ onwards until the end of the video sequence. Note that t is a per-frame index in the video, while m is a per-action index. A fully supervised setting is then associated with a set of data $\mathcal{F} = \{(\mathbf{x}, \mathbf{c}, \mathbf{d})\}$. We also denote the action labels and duration jointly by $\mathbf{y} = \{y_1, \ldots y_m \ldots y_M\}$, where $y_m = (c_m, d_m)$, and distinguish the ground truth and the corresponding predictions as y_m and $\hat{y}_m$ respectively.

Under a weakly-supervised setting, we assume we are given the set $\mathcal{W} = \{(\mathbf{x}, \mathbf{c}')\}$, *i.e.* observed videos of T observed frames $\mathbf{x} = \{x_1, \ldots, x_t, \ldots, x_T\}$, along with the weak label $\mathbf{c}' = c_1$, *i.e.* the action label of frame x_{T+1}. There are no assumptions on T, *i.e.* if the observed sequence end in the middle of an action, c_1 will be the current action label; if T is exactly the last frame of an action, then c_1 will be the label of the next action. This translates to the dense anticipation protocol of previous works [14, 28, 43] in which the first $X\%$ of a video is observed and predictions are made on the following $Y\%$ from X to $X+Y$. Therefore, we use as the weak label the first frame of the remaining $Y\%$.

We formulate dense anticipation as a mixed classification and regression task to anticipate the action labels and duration respectively. Without any assumption on the backbone anticipation method, we will refer to the primary module as function $f(\mathbf{x})$ and the conditional module as function $f_{\text{cond}}(\mathbf{x}, \mathbf{c}')$. The conditional module is trained based on the loss in Eq. 1. Then, $\mathcal{F}$ and $\mathcal{W}$ are used to train the primary module with conditional module fixed as elaborated in Section 3.3. The main issue is how to adjust pseudo-labels.

3.2 Conditional Module

The conditional module $\tilde{\mathbf{y}} = f_{\text{cond}}(\mathbf{x}, \mathbf{c}')$ is an auxiliary component trained for generating pseudo labels $\tilde{\mathbf{y}}$ for the weak set $\mathcal{W}$. To do so, it is trained in the standard way using $\mathcal{F}$ with the following loss function

$$L_{\text{cond}} = \frac{1}{|\mathcal{F}|} \sum_{\mathcal{F}} \sum_{m=1}^{M} \left(-c_m \log(\hat{c}_m^{\text{cond}}) + (d_m - \hat{d}_m^{\text{cond}})^2 \right), \tag{1}$$

where the first term is a cross-entropy loss for the anticipated action label $\hat{c}_m^{\text{cond}}$, while the second term is an MSE for the predicted action duration $\hat{d}_m^{\text{cond}}$. After training, the conditional module remains fixed. For generating pseudo labels, we simply apply f_{cond}. However, to make full use of the weak label, we replace the estimated $\hat{c}_1$ with the weak label $\mathbf{c}' = c_1$, *i.e.* $\tilde{\mathbf{y}} = \{(c_1, \hat{d}_1^{\text{cond}}), (\hat{c}_2^{\text{cond}}, \hat{d}_2^{\text{cond}}), \ldots, (\hat{c}_M^{\text{cond}}, \hat{d}_M^{\text{cond}})\}$.

3.3 Primary model

The primary module $\hat{\mathbf{y}} = f(\mathbf{x})$ predicts the future action and duration sequence $\hat{\mathbf{y}}$ given video $\mathbf{x}$ and is the module used for inference. During training, the objective is to minimize a loss

based on the labelled ground truth $\mathcal{F}$ and the refined pseudo-labels of $\mathcal{W}$, *i.e.*

$$\begin{aligned} L_{\text{prim}} = &\frac{1}{|\mathcal{F}|}\sum_{\mathcal{F}}\sum_{m=1}^{M}\left(-c_m\log(\hat{c}_m)+(d_m-\hat{d}_m)^2\right)+\frac{1}{|\mathcal{W}|}\sum_{\mathcal{W}}(-c_1\log(\hat{c}_1)) \\ &+\frac{1}{|\mathcal{W}|}\sum_{\mathcal{W}}\left(\sum_{m=2}^{M}\left(-\tilde{c}'_m\log(\hat{c}_m)\right)+\sum_{m=1}^{M}(\tilde{d}'_m-\hat{d}_m)^2\right), \end{aligned} \tag{2}$$

where $\hat{y}_m = (\hat{c}_m, \hat{d}_m)$ is the predicted label from the primary module while $\tilde{y}'_m = (\tilde{c}'_m, \tilde{d}'_m)$ is the refined pseudo-labels (see Sec. 3.4) on $\mathcal{W}$. The first two terms represent the loss based on ground truth labels on $\mathcal{F}$ and $\mathcal{W}$; we term this L_{label}. The third term in the loss is based on pseudo-labels on $\mathcal{W}$ and we term this $L_{\text{pseudo-label}}$.

3.4 Sequence Refinement

Directly using the pseudo-labels from the conditional module to train the primary module does not allow us to fully benefit from $\mathcal{W}$, since the conditional module is trained only on $\mathcal{F}$. As $\mathcal{F}$ is quite small (5-15% of the training set in our case), there is also the risk of confirmation bias [10]. To mitigate this possibility, we learn a refinement module to refine the pseudo-labels from the conditional module. For a video $\mathbf{x}$, the refinement module can be expressed as a function F applied the predicted labels from the primary module and the estimated pseudo-labels from the conditional module, *i.e.*

$$\tilde{\mathbf{y}}' = F(\hat{\mathbf{y}}, \tilde{\mathbf{y}}) = F\left(f(\mathbf{x}), f_{\text{cond}}((\mathbf{x}, \mathbf{c}'))\right). \tag{3}$$

We propose two refinement schemes as different options for F which we outline below.

Linear Refinement. As a naive baseline, we first propose to use a weighted geometric mean of the primary and conditional module outputs, where we consider $\hat{c}$ as a probability estimate over the classes. To that end, the refined label can be defined as

$$\tilde{\mathbf{y}}' = f(\mathbf{x})^{\frac{1}{\alpha+1}} \cdot f_{\text{cond}}((\mathbf{x}, \mathbf{c}'))^{\frac{\alpha}{\alpha+1}}, \tag{4}$$

where α is a hyperparameter determining the weighting of each component. Note that $\tilde{\mathbf{y}}'$ is actually the optimal solution when considering a linear weighting of the minimal KL divergences between (1) the refined pseudo-label $\tilde{\mathbf{y}}'$ and the estimate of the primary module $\hat{\mathbf{y}}$ as well as between $\tilde{\mathbf{y}}'$ and (2) the estimate of the conditional module $\tilde{\mathbf{y}}$. Intuitively, the refined output is the "closest" sequence to both modules' predictions.

From Eq. 4, it can be observed that when $\alpha = \infty$, $\tilde{\mathbf{y}}' = f_{\text{cond}}(\mathbf{x}, \mathbf{c}')$ while $\alpha = 0$ gives $\tilde{\mathbf{y}}' = f(\mathbf{x})$. These two extreme cases correspond to the refinement directly using the conditional or primary module outputs as the refined sequence respectively. We define a schedule for α to decrease from a large to a small value. This is based on the rationale that at the outset of training, the primary module is not so accurate and will need to rely on the conditional module, but as training progresses a smaller α is more suitable.

Adaptive Refinement. Instead of a manually set schedule for α, we can also directly learn a refined output. Ideally, we would like for the refined outputs $\tilde{\mathbf{y}}'$ to be more accurate than the outputs of both $f(\mathbf{x})$ and $f_{\text{cond}}(\mathbf{x}, \mathbf{c}')$. We can do this by leveraging the ground truth labels of

$\mathcal{F}$ and adding a loss on the refined output $\tilde{\mathbf{y}}'$:

$$L_{\text{adap}} = L_{\text{prim}} + \frac{1}{|\mathcal{F}|}\sum_{\mathcal{F}}\sum_{m=1}^{M}\Big(-c_m \log(\tilde{c}'_m) + (d_m - \tilde{d}'_m)^2\Big). \tag{5}$$

The adaptive refinement is realized via a linear layer that takes predicted and pseudo sequences and outputs a refined one. One key change made when learning the adaptive refinement as opposed to the linear refinement is that the conditional module is trained on only a portion of $\mathcal{F}$ (we opt for half out of simplicity). We purposely limit the training of the conditional module to prevent the refinement module from fully relying on its output. Then, $\mathcal{F}$ is used to train the primary and refinement module simultaneously. The objective function contains two parts: loss between output from the primary module $\hat{y}$ and ground truth (*i.e.* the first term in Eq. 2) and refined output $\tilde{y}'$ and ground truth (*i.e.* the second term in Eq. 5). Lastly, $\mathcal{F}$ as well as $\mathcal{W}$ is then applied to learn the primary and refinement module concurrently based on the loss in Eq. 5. We refer readers to Supplementary Section 8 to get a better idea of the training process.

3.5 Duration Attention

We introduce attention for the duration estimation; this is applicable only to recursive dense anticipation methods [14, 43]. At the decoder, the action label and duration for action m would be classified and regressed directly from the hidden state H_m. We propose to add an attention score between the hidden state and the input video to improve the duration estimate. Specifically, given video encoding $\mathcal{I}$, the attention weighted sum of the encoding can be defined as:

$$\textbf{attn}(H'_m, \mathcal{I}) = \text{softmax}(\frac{H'_m \mathcal{I}^\intercal}{\sqrt{d_I}})\mathcal{I}, \qquad \text{where} \qquad H'_m = WH_m + b \tag{6}$$

where $W \in \mathbb{R}^{d_I \times d_h}$ and $b \in \mathbb{R}^{d_I}$ are learned parameters, $\mathcal{I}^\intercal$ is the transpose of $\mathcal{I}$, d_h and d_I are the dimensionality of H_m and $\mathcal{I}$ respectively. The attention-based duration $\hat{d}_m$ is estimated as a linear transformation of the previous hidden state H_{m-1} and the weighted encoding:

$$\hat{d}_m = [\textbf{attn}(H'_m, \mathcal{I}), H_{m-1}]\beta + \varepsilon \tag{7}$$

where β, ε are learned parameters and $[\cdot]$ denotes a concatenation.

Duration Attention Regularizer. To further minimize the prediction differences between the primary and conditional module, we encourage the attention score between the two modules to be similar. To that end, we add to the objective functions Eq. 2 and Eq. 5 an l_2-norm between the attention scores of the conditional and primary modules, *i.e.*

$$L'_{\text{prim}} = L_{\text{prim}} + \sum_{m=1}^{M} \|\textbf{attn}_m^{\text{prim}} - \textbf{attn}_m^{\text{cond}}\|_2^2 \tag{8}$$

where $\textbf{attn}_m^{\text{prim}}$ and $\textbf{attn}_m^{\text{cond}}$ represent the attention scores of step m in the primary and conditional modules respectively. The same regularizer is also added to Eq. 5 to yield L'_{adap}.

4 Experiments

4.1 Datasets, Evaluation & Implementation Details

We evaluate our method on the two benchmark datasets used in dense anticipation: Breakfast Actions [16] and 50Salads [32]. Both datasets record realistic cooking activities, with each video featuring a sequence of continuous actions in making either a breakfast item or a salad[2]. From the designated training splits of each dataset, we partition 15% and 20% of the training data for the fully labelled set $\mathcal{F}$ for Breakfast and 50Salads respectively[3]. The remaining 85% / 80% of training sequences are assigned to $\mathcal{W}$ and have only a weak label, *i.e.* the single action label c_1 (see Sec. 3.1). Following the conventions of [14, 28, 43], we observe 20% or 30% of the video and anticipate the subsequent 20% and 50% of the video sequence (with additional results on 10% and 30% in the Supplementary Section 2). In line with previous works, we evaluate our anticipation results with mean over classes (MoC) [43].

As input features, we use the 64-dimension Fisher vectors computed on top of improved dense trajectories [17] as provided by [43] on a per-frame basis.

Currently, as all dense anticipation methods are fully supervised, there are no direct comparisons to competing state-of-the-art methods. However, as our framework is general, we experiment with 3 different anticipation methods as backbones in a series of self-comparisons. We test using (1) a naive RNN where both encoder and decoder is a one-layer LSTM with 512 hidden dimensions (2) the one-shot method of Ke [28] and (3) the recursive method of Sener [14]. Our result for Ke *et al.* is our re-implementation as they do not provide source code; our fully-supervised re-implementation yields similar values as their reported results. All hyperparameters follow the original settings in their papers.

For the linear refinement method, α begins from 30 and decreases to 0.5 with a decay rate of 0.95 per epoch. The batch size is 2 for 50Salads and 16 for Breakfast. Using linear refinement, the model converges at about 20 epochs for the first step and 25 epochs for the second step. The model converges at about 15 epochs for the first step, 20 for the second and third step when using adaptive refinement.

4.2 Supervised Baselines

We first compare the impact that the amount of data would have on the fully supervised case (see Table 1). We design three baselines and in each case, train a stand-alone primary module. Baseline (1) is fully supervised on the entire training set – this signifies the upper bound that our weakly-supervised method can achieve. Baseline (2) is supervised on only the labelled set $\mathcal{F}$. This baseline gives some indicator of the accuracy of the conditional module before the weak label is applied to replace $\hat{c}_1$ and acts as a lower bound. Baseline (3) supervised on the given labels of $\mathcal{F}$ and $\mathcal{W}$, *i.e.* applying the first two terms or L_{label} of Eq. 2. This baseline tells us what can be learned from the full set of provided labels.

Full supervision with the entire training set, *i.e.* Baseline (1) achieves the best results, with the model of Sener *et al.* [14] performing best. However, performance drops with fewer labels, *i.e.* Baselines (2) and (3) and the one-shot method of Ke *et al.* [28] is slightly stronger than [14]. The gains from adding the labels of the weak set $\mathcal{W}$, *i.e.* from Baseline (2) to (3) demonstrate that having even a single c_1 label helps to improve MoC by 1-2%.

[2] Dataset details are in the Supplementary Section 1.

[3] We use a slightly higher percentage for 50Salads due to the small dataset size

Table 1: MoC of different models. Results reported in Baseline (1) for Ke [28] and Sener [14] are taken directly from their published results. Other results are averaged on the officially provided different splits for training (which is further split into fully- and weakly-labeled sets randomly according to the percentages mentioned above) and test set.

	Breakfast				50Salads			
Obs.	20%		30%		20%		30%	
Pred.	20%	50%	20%	50%	20%	50%	20%	50%
Baseline 1: $f(\mathbf{x})$, fully-supervised on entire training set (theoretical upper bound)								
RNN	6.53	5.30	8.52	5.37	9.71	7.82	12.64	8.54
Ke [28]	11.92	7.03	12.26	8.18	11.53	9.50	15.92	9.89
Sener [14]	13.10	11.10	17.00	15.10	19.90	15.10	22.50	11.20
Baseline 2: $f(\mathbf{x})$, supervised on full label set $\mathcal{F}$ (theoretical lower bound)								
RNN	3.92	2.35	5.48	4.26	8.08	5.45	8.13	6.70
Ke [28]	6.81	5.39	7.32	5.88	8.36	4.51	11.19	8.23
Sener [14]	6.19	4.90	7.30	5.92	8.67	7.01	12.73	8.00
Baseline 3: $f(\mathbf{x})$, supervised on full label set $\mathcal{F}$ + weak set $\mathcal{W}$ with L_{label}								
RNN	6.01	4.29	7.56	5.93	9.33	6.96	11.45	8.54
Ke [28]	8.89	5.71	10.05	7.59	9.25	6.11	13.17	9.80
Sener [14]	7.64	5.54	8.05	6.77	9.97	7.89	13.30	9.61
Our model with adaptive refinement but without duration attention.								
RNN	7.85	7.96	8.33	8.21	10.48	7.40	13.04	10.05
Ke [28]	9.74	6.24	11.02	9.24	11.84	9.27	13.88	12.81
Sener [14]	8.98	7.71	9.71	7.31	12.62	9.44	13.94	10.73
Our full model with adaptive refinement and duration attention.								
RNN	9.12	8.33	10.17	8.90	12.11	9.57	14.37	10.91
Sener [14]	9.74	8.56	11.63	8.99	12.41	9.67	14.94	12.14

4.3 Impact of Adding Pseudo-Labels and Duration Attention

If we add pseudo-labels to train the primary module, *i.e.* by applying the full loss given in Eq. 2 (see Table 1, fourth section) and using adaptive refinement, we observe that we gain in performance across the board when compared to Baseline (3), even though it uses the same amount of provided ground truth labels. The most impressive is the RNN encoder-decoder model. With only the pseudo-labels from the weak set, we can surpass the original fully supervised baseline. Using the one-shot method from Ke [28], we can surpass the supervised baseline when anticipating 50% of the sequence after observing 30% for both Breakfast and 50Salads. On Sener's model [14], however, we are not able to surpass the fully supervised baseline, though the gap closes progressively. Our full model (Table 1, fifth section) which incorporates the attention duration sees additional gains in most settings. There is also a visual explanation in Supplementary Section 4 which intuitively illustrates different correlations between different observed actions and current predicted action. Note that we do not apply the duration attention to the model of Ke [28] since it is not recursive.

All three backbones improve from Baseline (3) when adding adaptive refinement and duration attention. Given the challenge of the dense anticipation task, however, the overall performance is still very low, especially for the simple RNN and Ke's [28] model. This is likely the reason why adding our framework can outperform the fully supervised case. As the models are rather simplistic, we speculate they cannot fully leverage all the ground truth labels from the entire training dataset (Table 1, Baseline (1)). Training with our framework (Table 1, our model in purple and white section) may result in even higher accuracies because

our refined pseudo-labels, while less accurate than ground truth, model a simpler distribution.

4.4 Future Horizon of Anticipated Actions

We analyze in Table 2 the anticipated actions over time by computing the accuracy for the first future action (weak label) versus the next three actions (no label). The trends for the two settings are very different; Baseline 3 without the conditional module has a sharp drop-off from the second action. This is unsurprising since most videos have only a weak label of the first future action. Incorporating our conditional module with the refined pseudo-labels improves the first action's accuracy and decreases the drop-off of subsequent actions. Refer to Supplementary Section 7 for a visualization of the anticipated action sequence.

Table 2: Accuracy of the predicted actions at different time steps.

	First	Second	Third	Fourth
Baseline 3	16.17	6.49	3.22	1.67
Our full model	18.75	14.33	9.09	5.49

4.5 Ablation Study

In the following experiments we use Sener's [14] method as the backbone, an observation of 30% and anticipation of 10%. Table 3 verifies that refining the pseudo-labels is more effective than training with them directly. Furthermore, the learned adaptive refinement is better than the linear refinement as it improves upon the linear scheme by 4% on both datasets.

In addition to Fisher vector IDT features, we also experiment with the ground truth labels and the stronger I3D features [2] as inputs, the result is shown in Table 4. To use ground truth labels as input, we simply use a one-hot vector. It gives much higher accuracy, indicating that there is still some gap in recognition performance. The same gap was also confirmed in [14]. In line with previous results which use both features, I3D achieves higher MoC than Fisher vector. We observe, however, that using I3D features requires longer training time, *i.e.* 20 epochs in step 1 and 2, 25 epochs in step 3 (we refer readers to Section 3.4 in the main paper and Section 8 in the Supplementary for a detailed training procedure), likely due to the larger dimensionality of I3D compared to Fisher vectors.

Table 3: MoC on different refinements.

	Breakfast	50Salads
No refinement	6.28	10.31
Linear	7.79	12.17
Adaptive	12.78	16.24

Table 4: MoC on different video features.

	Breakfast	50Salads
Ground truth	61.30	35.40
Fisher vector	12.78	16.24
I3D	15.65	21.30

5 Conclusion

In this paper, we investigate a novel dense anticipation task, emphasizing pseudo labels to promote anticipation accuracy using weakly-labelled videos. To predict accurate action/duration sequences, we propose a sequence refinement method that generates pseudo sequences conditioned on the next-step action and adaptively refines the pseudo sequences to guide prediction. We also introduce duration attention which takes action correlations into account to boost duration anticipation. The proposed method outperforms, if not better than, other fully supervised methods while requiring far less annotation effort. More datasets will be involved in future works.

Acknowledgements This research is supported by the National Research Foundation, Singapore under its NRF Fellowship for AI (NRF-NRFFAI1-2019-0001).

References

[1] A.Furnari and G.Farinella. Rolling-unrolling lstms for action anticipation from first-person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4021–4036, 2020.

[2] A.Iscen, G.Tolias, Y.Avrithis, and O.Chum. Label propagation for deep semi-supervised learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] A.Tarvainen and H.Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint*, page arXiv:1703.01780, 2017.

[4] C.Canuto, P.Moreno, J.Samatelo, R.Vassallo, and J.Santos-Victor. Action anticipation for collaborative environments: The impact of contextual information and uncertainty-based prediction. *Neurocomputing*, 444, 2020.

[5] C.Vondrick, H.Pirsiavash, and A.Torralba. Anticipating visual representations from unlabeled video. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] C.Wang, H.Gu, and W.Su. Sar image classification using contrastive learning and pseudo-labels with limited data. *IEEE Geoscience and Remote Sensing Letters*, Early Access:1–5, 2021.

[7] D.Berthelot, N.Carlini, I.Goodfellow, N.Papernot, A.Oliver, and C.Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[8] D.Damen, H.Doughty, G.Farinella, S.Fidler, A.Furnari, E.Kazakos, D.Moltisanti, J.Munro, T.Perrett, W.Price, and M.Wray. Scaling egocentric vision: The epic-kitchens dataset. *European Conference on Computer Vision (ECCV)*, 2018.

[9] D.Moltisanti, F.Sanja, and D.Dima. Action recognition from single timestamp supervision in untrimmed videos. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] E.Arazo, D.Ortego, P.Albert, N. E. O'Connor, and K.McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *International Joint Conference on Neural Networks (IJCNN)*, 2020.

[11] E.Dessalene, C.Devaraj, M.Maynord, C.Fermuller, and Y.Aloimonos. Forecasting action through contact representations from first person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Early Access:1–1, 2021.

[12] F.Ma, L.Zhu, Y.Yang, S.Zha, G.Kundu, M.Feiszli, and A.Shou. Sf-net: Single-frame supervision for temporal action localization. *European Conference on Computer Vision (ECCV)*, 2020.

[13] F.Pirri, L.Mauro, E.Alati, V.Ntouskos, M.Izadpanahkakhk, and E.Omrani. Anticipation and next action forecasting in video: an end-to-end model with memory. *arXiv preprint*, page arXiv:1901.03728, 2019.

[14] F.Sener, D.Singhania, and A.Yao. Temporal aggregate representations for long-range video understanding. *European Conference on Computer Vision (ECCV)*, 2020.

[15] H.Koppula and A.Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2015.

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

[17] H.Wang and C.Schmid. Action recognition with improved trajectories. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[18] H.Wu and S.Prasad. Semi-supervised deep learning using pseudo labels for hyperspectral image classification. *IEEE Transactions on Image Processing*, 27(3):1259 – 1270, 2017.

[19] H.Yu and W.Zheng. Weakly supervised discriminative feature learning with state information for person identification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] H.Zhang, M.Cisse, Y.Dauphin, and D.Lopez-Paz. Mixup: Beyond empirical risk minimization. *arXiv preprint*, page arXiv:1710.09412, 2017.

[21] J.Ahn, S.Cho, and S.Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[22] J.Carreira and A.Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] J.Dong, Y.Cong, G.Sun, and D.Hou. Semantic-transferable weakly-supervised endoscopic lesions segmentation. *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[24] J.Lee, E.Kim, and S.Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[25] J.Wang, C.Ding, S.Chen, C.He, and B.Luo. Semi-supervised remote sensing image semantic segmentation via consistency regularization and average update of pseudo-label. *Remote Sensing*, 12(21):3603, 2020.

[26] L.Chen, W.Wu, C.Fu, X.Han, and Y.Zhang. Weakly supervised semantic segmentation with boundary exploration. *European Conference on Computer Vision (ECCV)*, 2020.

[27] M.Liu, S.Tang, Y.Li, and J.M.Rehg. Forecasting human-object interaction: joint prediction of motor attention and actions in first person video. *European Conference on Computer Vision (ECCV)*, 2020.

[28] Q.Ke, M.Fritz, and B.Schiele. Time-conditioned action anticipation in one shot. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[29] S.Helmstetter and H.Paulheim. Weakly supervised learning for fake news detection on twitter. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2018.

[30] S.Laine and T.Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint*, page arXiv:1610.02242, 2016.

[31] S.Qi, S.Huang, P.Wei, and S.Zhu. Predicting human activities using stochastic grammar. *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[32] S.Stein and S.McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2013.

[33] T.Lan, T.Chen, and S.Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision (ECCV)*. Springer, 2014.

[34] T.Mahmud, M.Hasan, and A.K.Roy-Chowdhury. Joint prediction of activity labels and starting times in untrimmed videos. *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[35] T.Zhang, W.Min, Y.Zhu, Y.Rui, and S.Jiang. An egocentric action anticipation framework via fusing intuition and analysis. *ACM International Conference on Multimedia*, 2020.

[36] W.Ge, X.Lin, and Y.Yu. Weakly supervised complementary parts models for fine-grained image classification from the bottom up. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] W.Wang, X.Peng, Y.Su, Y.Qiao, and J.Cheng. Ttpp: Temporal transformer with progressive prediction for efficient action anticipation. *Neurocomputing*, 438:270–279, 2021.

[38] W.Yang, T.Zhang, X.Yu, T.Qi, Y.Zhang, and F.Wu. Uncertainty guided collaborative training for weakly supervised temporal action detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[39] X.Zhang, Z.Peng, P.Zhu, T.Zhang, C.Li, H.Zhou, and L.Jiao. Adaptive affinity loss and erroneous pseudo-label refinement for weakly supervised semantic segmentation. *arXiv preprint*, page arXiv:2108.01344, 2021.

[40] Y.Chang, Q.Wang Q, W.Hung, R.Piramuthu, Y.Tsai YH, and M.Yang. Weakly-supervised semantic segmentation via sub-category exploration. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[41] Y.Chang, Q.Wang, W.Hung, R.Piramuthu, Y.Tsai, and M.Yang. Weakly-supervised semantic segmentation via sub-category exploration. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[42] Y.Chen, X.Zhu, and S.Gong. Semi-supervised deep learning with memory. *European Conference on Computer Vision (ECCV)*, 2018.

[43] Y.Farha, A.Richard, and J.Gall. When will you do what? - anticipating temporal occurrences of activities. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[44] Y.Liu, Y.Wu, P.Wen, Y.Shi, Y.Qiu, and M.Cheng. Leveraging instance-, image- and dataset-level information for weakly supervised instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Early Access:1–1, 2020.

[45] Y.Meng, J.Shen, C.Zhang, and J.Han. Weakly-supervised neural text classification. *ACM International Conference on Information and Knowledge Management (CIKM)*, 2018.

[46] Y.Ng and F.Basura. Forecasting future action sequences with attention: a new approach to weakly supervised action forecasting. *IEEE Transactions on Image Processing*, 29:8880–8891, 2020.

[47] Y.Yao, T.Chen, G.Xie, C.Zhang, F.Shen, Q.Wu, Z.Tang, and J.Zhang. Non-salient region object mining for weakly supervised semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[48] Z.Fang, G.Zhang, Q.Dai, Y.Kong, and P.Wang. Semisupervised deep convolutional neural networks using pseudo labels for polsar image classification. *IEEE Geoscience and Remote Sensing Letters*, Early Access:1–5, 2020.

[49] Z.Li, Y.Farha, and J.Gall. Temporal action segmentation from timestamp supervision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.