

Supplementary: Weakly-Supervised Dense Action Anticipation

Haotong Zhang¹
haotongz@u.nus.edu

Fuhai Chen²
cfh3c@nus.edu.sg

Angela Yao²
ayao@comp.nus.edu.sg

¹ Department of Mathematics
National University of Singapore
Singapore

² Department of Computer Science
National University of Singapore
Singapore

1 Dataset Details

Breakfast Actions [1] contains 1712 videos which are performed by 52 different individuals in 18 different kitchens. The videos are unscripted and uncontrolled with natural lighting, view points and environments. **50Salads** [2] is food preparation dataset capturing 25 people preparing 2 mixed salads each. Both datasets have standardized train-test splits which we follow. We further split the training set into fully- and weakly-labelled sets, with specific proportions and other details in Table 1.

Table 1: Basic information of dataset.

Dataset	fps	Video duration median, mean±std	Classes	Total	Train	Full	Weak	Test
Breakfast	15	91s, 140s±122	48	1712	1460	15%	85%	252
50Salads	30	389s, 370s±106	19	50	40	20%	80%	10

2 Complete Results

We provide a complete set of anticipations (10%, 20%, 30% and 50%) in Tables 2 and 3 for Breakfast and 50Salads respectively. Findings are consistent with the 20% and 50% results in the main paper. Baseline 1 is a fully supervised version; the MoC of Baseline 2 drops because we omit a large proportion of videos from the training set. We observe an increase in the performance of Baseline 3 compared to Baseline 2 when weak labels are added back to help training. The boosts manifested in Baseline 4 and 5 indicate the advantage of pseudo labels and duration attention respectively.

Table 2: MoC of different methods on Breakfast. Better viewed in colour.

Observed	20%				30%			
Predicted	10%	20%	30%	50%	10%	20%	30%	50%
Baseline 1: $f(\mathbf{x})$, fully-supervised on entire training set (theoretical upper bound)								
RNN	8.39	6.53	5.93	5.30	9.19	8.52	7.92	5.37
Ke [■]	13.04	11.92	7.76	7.03	14.24	12.26	11.60	8.18
Sener [■]	15.60	13.10	12.10	11.10	19.50	17.00	15.60	15.10
fully-supervised on the fully labelled subset (theoretical lower bound)								
RNN	5.48	3.92	3.45	2.35	5.98	5.48	5.23	4.26
Ke [■]	7.18	6.81	5.32	5.39	9.83	7.32	6.33	5.88
Sener [■]	7.47	6.19	5.18	4.90	7.93	7.30	5.47	5.92
Baseline 3: $f(\mathbf{x})$, supervised on full label set \mathcal{F} + weak set \mathcal{W} with L_{label}								
RNN	7.29	6.01	5.16	4.29	8.34	7.56	6.62	5.93
Ke [■]	9.76	8.89	6.51	5.71	11.71	10.05	8.52	7.59
Sener [■]	8.09	7.64	6.37	5.54	9.38	8.05	7.45	6.77
Our model with adaptive refinement but without duration attention.								
RNN	9.87	7.85	6.89	7.96	10.90	8.33	8.31	8.21
Ke [■]	11.82	9.74	7.32	6.24	13.75	11.02	10.06	9.24
Sener [■]	9.03	8.98	7.64	7.71	10.11	9.71	8.11	7.31
Our full model with adaptive refinement and duration attention								
RNN	9.93	9.12	8.70	8.33	12.55	10.17	9.54	8.90
ours	10.09	9.74	7.99	8.56	12.78	11.63	10.73	8.99

Table 3: MoC of different methods on 50Salads. Better viewed in colour.

Observed	20%				30%			
Predicted	10%	20%	30%	50%	10%	20%	30%	50%
Baseline 1: $f(\mathbf{x})$, fully-supervised on entire training set (theoretical upper bound)								
RNN	11.49	9.71	9.60	7.82	12.97	12.64	11.83	8.54
Ke [■]	12.29	11.53	10.97	9.50	16.34	15.92	11.56	9.89
Sener [■]	25.50	19.90	18.20	15.10	30.60	22.50	19.10	11.20
Baseline 2: $f(\mathbf{x})$, supervised on full label set \mathcal{F} (theoretical lower bound)								
RNN	9.81	8.08	6.59	5.45	10.65	8.13	7.52	6.70
Ke [■]	9.16	8.36	7.65	4.51	12.69	11.19	8.31	8.23
Sener [■]	11.36	8.67	7.30	7.01	13.16	12.73	10.95	8.00
Baseline 3: $f(\mathbf{x})$, supervised on full label set \mathcal{F} + weak set \mathcal{W} with L_{label}								
RNN	10.60	9.33	8.31	6.96	13.25	11.45	10.55	8.54
Ke [■]	11.87	9.25	8.83	6.11	14.97	13.17	10.74	9.80
Sener [■]	12.91	9.97	8.86	7.89	14.63	13.30	11.19	9.61
Our model with adaptive refinement but without duration attention.								
RNN	12.72	10.48	9.84	7.40	14.52	13.04	12.72	10.05
Ke [■]	15.00	11.84	10.96	9.27	15.66	13.88	12.89	12.81
Sener [■]	13.07	12.62	10.01	9.44	15.25	13.94	11.44	10.73
Our full model with adaptive refinement and duration attention								
RNN	14.53	12.11	10.06	9.57	15.09	14.37	13.25	10.91
ours	16.80	12.41	10.12	9.67	16.24	14.94	13.53	12.14

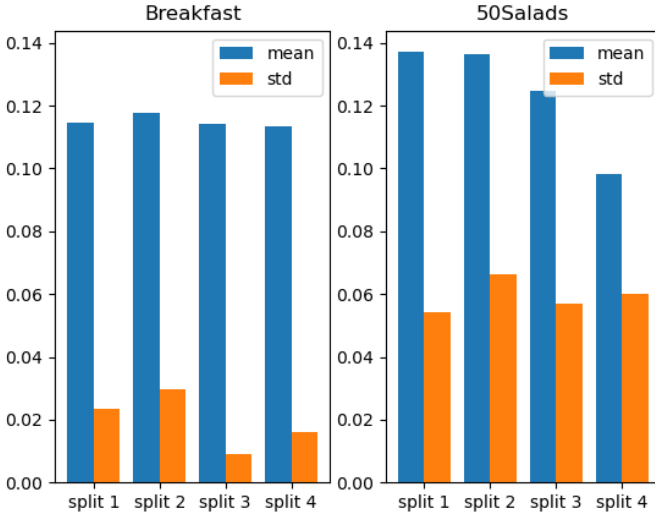


Figure 1: Bar charts of means and standard deviations in each split.

3 Variance in One Split

To further prove the randomness of our data choice and observe the variance in one split, we run 10 times on each split and plot the means and standard deviations. Here we use observation of 30% and prediction of 20% and use Sener’s [10] method as the backbone. As shown in Figure 1, we can see standard deviations of 50Salads are higher than those of Breakfast. The Reason may be that 50Salads has fewer videos, which is more unstable.

4 Visualization of Attention Scheme

We use a heat map (Figure 2) to further illustrate the advantage of our duration attention scheme. Take a video in 50Salads as an example, we track the attention score between current predicted action and observed actions. In the heat map, it’s obvious that the correlation between “cut cucumber” and “peel cucumber” as well as the correlation between “place tomato into bowl” and “cut tomato” are the highest, which indicates that more relevant actions have more influence on current action duration.

5 Full-Weak Split

We vary the proportion of fully-labelled data in the training set in Table 4 and observe that by increasing the proportion of fully-labelled data (the total amount of data is fixed), the performance gets progressively closer to the fully supervised model. For the RNN and Ke’s [9] model, we are able to exceed the performance of the fully supervised model, though this is largely due to their poor baseline performance even with 100% of the training data fully

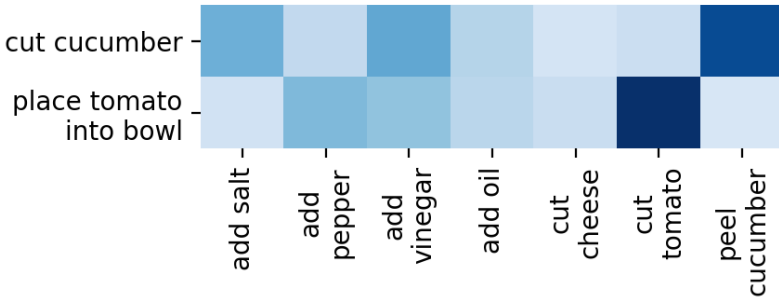


Figure 2: Heat map of attention score between current predicted action and observed actions. x-axis is observed actions and y-axis is predicted actions. Deeper colour indicates higher attention score.

supervised. It is likely that these models, being simpler and having fewer parameters, require a smaller proportion of fully-labelled data. For a larger model like that of Sener [14], 25% / 30% of the data is not sufficient to match the fully-supervised performance.

We omit experiments for RNN with split 25% on Breakfast and split 30% on 50Salads because MoC with smaller splits already exceeds fully-supervised results.

Table 4: MoC on different full vs. weak data splits. Percentages indicate the proportion of fully-labelled data in the training set. RNN and Sener [14] use our full model in the weakly-supervised setting, *i.e.* with duration attention while Ke [15], as a one-shot method, does not have duration attention. *100% indicates the original fully-supervised model (also without duration attention).

	Breakfast				50Salads			
	5%	15%	25%	100%*	10%	20%	30%	100%*
RNN	11.03	12.55		9.19	8.84	15.09		12.97
Ke [15]	12.40	13.75	17.45	14.24	11.73	15.66	20.00	16.34
Sener [14]	11.90	12.78	17.22	19.50	14.63	16.24	17.37	30.60

6 Memory Complexity Analysis

A simple comparison of memory complexity (expressed by the number of hyperparameters) of three baseline models (with fully-supervised setting) and our full model with adaptive refinement is shown in Table 5. Not surprisingly, the more complicated model has more hyperparameters. The number of hyperparameters of our full model is approximately two times the corresponding backbone’s, which is in accordance with our intuition that the primary and conditional module is similar and are both based on the backbone. We omit time complexity analysis because it is not comparable between fully- and weakly-supervised models.

Table 5: Memory complexity analysis.

	RNN	Ke [10]	Sener [11]	Ours with RNN	Ours with Sener's
50Salads	3579944	8937316	36997429	7159894	75374068
Breakfast	3624546	9845839	40270080	7249098	81590790

7 Visualized Result

Figure 3 shows an example of anticipating 50% of the sequence after observing 30%, where each colour indicates an action. We can see that the action sequence is correct, but there are some errors in the predicted duration.

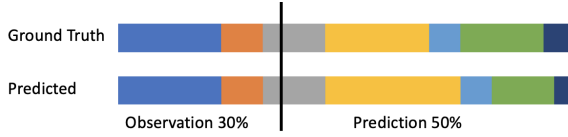


Figure 3: Visualized result for dense anticipation on Breakfast.

8 Pseudo Codes

Below are pseudo codes for linear refinement and adaptive refinement respectively.

Algorithm 1 Linear Refinement

Require: initial model $Prim, Cond$; \mathcal{W}, \mathcal{F} ; Epoch N_1, N_2 ; α , decay parameter d

Step 1:

for $n = 1$ to N_1 **do**

$pseudo_label \leftarrow Cond(\mathcal{F})$

$L \leftarrow Loss(ground_truth, pseudo_label)$

 Update $Cond$ by minimizing L

end for

Fix $Cond$

Step 2:

for $n = 1$ to N_2 **do**

$predicted_label \leftarrow Prim(\mathcal{F})$

$L_1 \leftarrow Loss(ground_truth, predicted_label)$

$predicted_label \leftarrow Prim(\mathcal{W})$

$pseudo_label \leftarrow Cond(\mathcal{W})$

$refined_label \leftarrow predicted_label^{\frac{1}{\alpha+1}} * pseudo_label^{\frac{\alpha}{\alpha+1}}$

$L_2 \leftarrow Loss(refined_label, predicted_label)$

 Update $Prim$ by minimizing $L_1 + L_2$

$\alpha \leftarrow d * \alpha$

end for

Algorithm 2 Adaptive Refinement**Require:** initial model $Prim, Cond, Refine$; \mathcal{W}, \mathcal{F} ; Epoch N_1, N_2, N_3

Step 1:

for $n = 1$ to N_1 **do** $pseudo_label \leftarrow Cond(\mathcal{F})$ $L \leftarrow Loss(ground_truth, pseudo_label)$ Update $Cond$ by minimizing L **end for**Fix $Cond$

Step 2:

for $n = 1$ to N_2 **do** $predicted_label \leftarrow Prim(\mathcal{F})$ $pseudo_label \leftarrow Cond(\mathcal{F})$ $refined_label \leftarrow Refine(predicted_label, pseudo_label)$ $L_1 \leftarrow Loss(ground_truth, predicted_label)$ $L_2 \leftarrow Loss(refined_label, predicted_label)$ Update $Prim, Refine$ by minimizing $L_1 + L_2$ **end for**

Step 3:

for $n = 1$ to N_3 **do** $predicted_label \leftarrow Prim(\mathcal{F})$ $pseudo_label \leftarrow Cond(\mathcal{F})$ $refined_label \leftarrow Refine(predicted_label, pseudo_label)$ $L_1 \leftarrow Loss(ground_truth, predicted_label)$ $L_2 \leftarrow Loss(refined_label, predicted_label)$ $predicted_label \leftarrow Prim(\mathcal{W})$ $pseudo_label \leftarrow Cond(\mathcal{W})$ $refined_label \leftarrow Refine(predicted_label, pseudo_label)$ $L_3 \leftarrow Loss(refined_label, predicted_label)$ Update $Prim, Refine$ by minimizing $L_1 + L_2 + L_3$ **end for****References**

- [1] F.Sener, D.Singhanian, and A.Yao. Temporal aggregate representations for long-range video understanding. In *ECCV*, 2020.
- [2] H.Kuehne, A.Arslan, and T.Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014.
- [3] Q.Ke, M.Fritz, and B.Schiele. Time-conditioned action anticipation in one shot. In *CVPR*, 2019.
- [4] S.Stein and S.McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *UbiComp*, 2013.