Supplementary Material for: Exemplar-Based Early Event Prediction in Video

Zekun Zhang[1]
zekzhang@cs.stonybrook.edu
Farrukh M. Koraishy[2]
farrukh.koraishy@stonybrookmedicine.edu
Minh Hoai[1]
minhhoai@cs.stonybrook.edu

[1] Department of Computer Science
[2] Division of Nephrology
Department of Medicine

Stony Brook University
New York, 11794, USA

Abstract

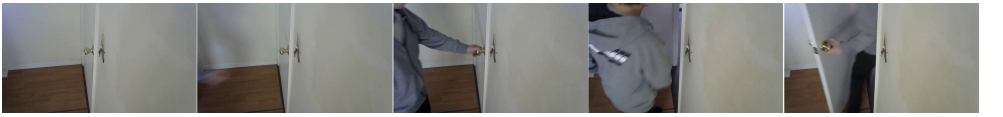
In this supplementary material, we show some additional figures and videos that cannot be fitted into the main paper. We also present some sample frames and detailed information of the datasets used in the paper. We do more ablation study to explore the hyper-parameters of the proposed model, and investigate the variance of the results. This document is best to be read on digital screen.

1 Sample frames from videos

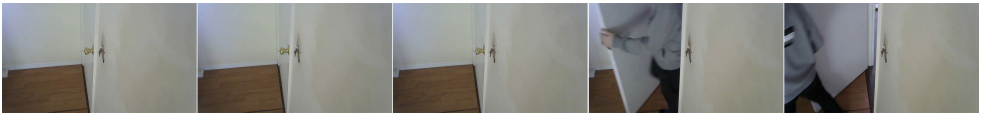
Here we show some sample frames taken from the datasets used in the experiments. Figure 1 shows two frame sequences from the OpenDoor video. Figure 1a shows the frame sequence of the door being opened from the inside, and Figure 1b shows the frame sequence of the door being opened from the outside. In Figure 2 two frame sequences from the PickupPhone video are shown. Figure 2a shows the frame sequence of the phone being picked up after a notification pops up on the screen, and Figure 2b shows the frame sequence of the person picking up the phone without any notification. Figure 3 shows a frame sequences taken from the EpicKitchen video `P01_01.MP4` around time `00:05:01`, when an *open-tap* action is being performed.

2 Video illustration of EnEx prediction scores

We visualize how the prediction scores produced by EnEx classifier changes over time when new frames in the video is observed on PickupPhone actions with lead time $\tau_1 = 0.5$, and OpenDoor events with lead time $\tau_1 = 2s$. We use the same sampling method discussed in the paper, and train EnEx model on the first half of the replayed samples using the same settings and hyper-parameters. Then the model is applied to the remaining samples for visualization. In the visualization stage, the score at time t is calculated as the EnEx outputs of the max-pooling frame-wise feature in the time window $[t-l,t]$, where the feature is extracted from pre-trained ResNet-34 network. This is essentially the same setting used in the paper, with



(a) door being opened from inside



(b) door being opened from outside

Figure 1: Sample frame sequences taken from OpenDoor video.

(a) No Notification

(b) No Notification

Figure 2: Sample frame sequences taken from PickupPhone video.

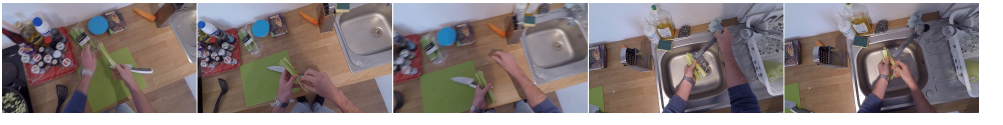


Figure 3: Sample frame sequences taken from EpicKitchen video P01_01.MP4.

the only difference that sliding window is applied to obtain how the prediction scores change continuously when every new frame is observed.

The results are illustrated in the attached video file `EnEx.mp4`. 4 video clips are shown along with the corresponding prediction scores. In the first clip, the phone gets picked up after the person sees a notification on the screen. It is clear that the prediction score increases immediately when the notification pops up, indicating that after training, EnEx can learn to predict with high confidence after seeing the precursory clue that will surely lead to the PickupPhone action. The second clip shows the phone gets picked up without any notification is seen. This time the action is not predictable, and the prediction score stays low until the action actually starts. This show that after training, EnEx can distinguish unpredictable actions from predictable ones. By doing this, it can make predictions with high precision.

The third clip shows the event of the door being opened from inside. In this case, the precursory clues are more subtle to observe. The video shows that when the person approaches the door, the lighting and shadow on the wall changes. The trained EnEx model can pick such clues, and produces higher prediction scores. The forth video clip shows the event of the

Table 1: Ablation study with different methods of calibration and combination

options		options used					
calibration	logistic		✓	✓		✓	
	ours	✓			✓		✓
combination	max	✓	✓	✓	✓		
	ours					✓	✓
classification	eSVM	✓	✓				
	EnEx			✓	✓	✓	✓
		36.3	53.2	66.3	36.5	76.9	**82.1**

door being opened from outside. The prediction score stays low until the event actually starts. Similar to the illustration for PickupPhone action, this shows that EnEx can make prediction with high precision when the event is predictable.

3 Ablation study on effectiveness of calibration and combination methods

We perform some ablation experiments to show the effectiveness of the proposed calibration and combination methods. For calibration, we can use the proposed non-parametric method or Platt scaling as in eSVM. For combination, we can either use the the proposed rank pooling or max pooling of eSVM. The results are shown in Table 1. The numbers are the running average of $AP@0.1$. EnEx works well with both calibration methods, while eSVM must be used with logistic calibration. Max pooling works much better if logistic calibration is used, but it is not as robust as the proposed combination method.

4 Calculation time of EnEx model

The proposed non-parametric calibration runs slightly faster than logistic calibration. EnEx training is also faster than eSVM. On a small dataset of 70 positive and 150 negative examples, the training (including calibration) time of EnEx and eSVM are 240ms and 540ms, respectively. On a bigger dataset with 200 positive and 860 negative examples, the training duration are 7.2s and 27.4s respectively. The inference time for one query are 0.4ms and 12ms respectively. All measurements are performed on a MacBook Pro with 2.6 GHz Quad-Core Intel Core i7 CPU.

5 Hyper-parameters for EnEx model

We did some exploration on the hyper-parameters of the proposed EnEx model. Those are the coefficient γ used in the RBF kernel, and the regularization strength λ. In experiments we set

$$\gamma = k_\gamma / \bar{d}, \tag{1}$$

Table 2: Results of different hyper-parameters combinations in EnEx model on predicting OpenDoor actions with lead time of 2s. The combination used in all other experiments is highlighted.

		k_γ				
		10^{-2}	10^{-1}	10^{0}	10^{1}	10^{2}
k_λ	10^{-4}	79.9	77.3	78.9	85.3	55.7
	10^{-3}	74.1	76.0	79.2	75.9	57.9
	10^{-2}	63.9	77.4	81.9	81.4	66.3

(a) Average $AP@0.1$

		k_γ				
		10^{-2}	10^{-1}	10^{0}	10^{1}	10^{2}
k_λ	10^{-4}	45.7	45.6	44.4	48.4	35.8
	10^{-3}	41.3	45.6	44.8	43.6	36.3
	10^{-2}	35.6	42.9	44.4	47.5	39.1

(b) Average $AP@1$

where $\bar{d}$ is the averaged Euclidean distance between the training samples; and

$$\lambda = k_\lambda (n_- + 1) \tag{2}$$

where n_- is the number of negative training samples. We try different combinations of k_γ and k_λ. and run each combinations to predict OpenDoor actions with lead time $\tau_1 = 2$s for 20 times. The average $AP@0.1$ and $AP@1$ are reported in Table 2. Note that in all other experiments, we have set $k_\gamma = 1$ and $k_\lambda = 10^{-3}$.

6 Variance of results

In the main paper, we only show the average result of experiment runs to avoid clutter. Here, we show the results together with the Standard Errors (SE), which is defined as

$$SE = \sqrt{\frac{1}{n(n-1)} \sum_{i=1}^{n} (x_i - \bar{x})^2}, \tag{3}$$

Where $\bar{x}$ is the average result of all experiment runs, and n is the number of experiment runs. The experiments are run for $n = 100$ times on PickupPhone events with $\tau_1 = 0.25$s. The standard error over different experiment runs at every training iterations is plotted in Figure 4 with filled error bars.

7 Final scores vs. average scores

In the main paper, we report the performance of models using the average scores over different iterations to mimic the online learning scheme. Here we also report the final scores, which is the AP scores at the end of the training, when the number of training samples is at maximum. Figure 5 shows both the final scores and average scores of different models under different recall thresholds as precision-recall curves. The models are trained to predict OpenDoor actions with different lead times. With each lead time, experiments are run for 10 times.

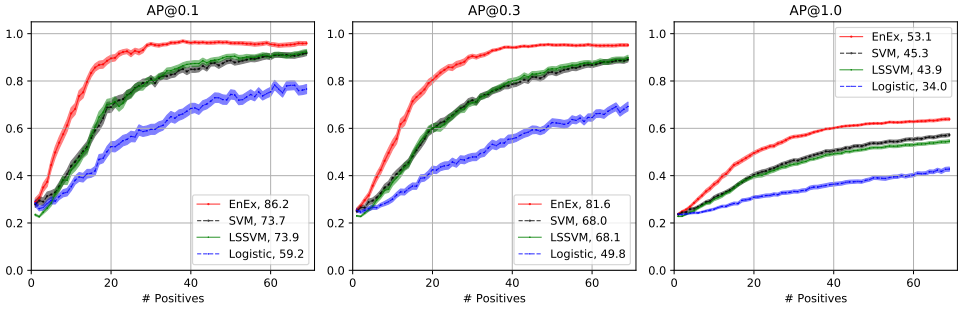
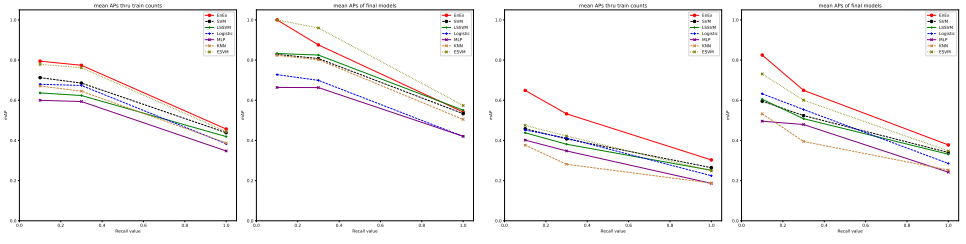


Figure 4: Standard error of results over different experiment runs at every training iterations on PickupPhone events with $\tau_l = 0.25$s. Standard error is plotted using filled error bars.

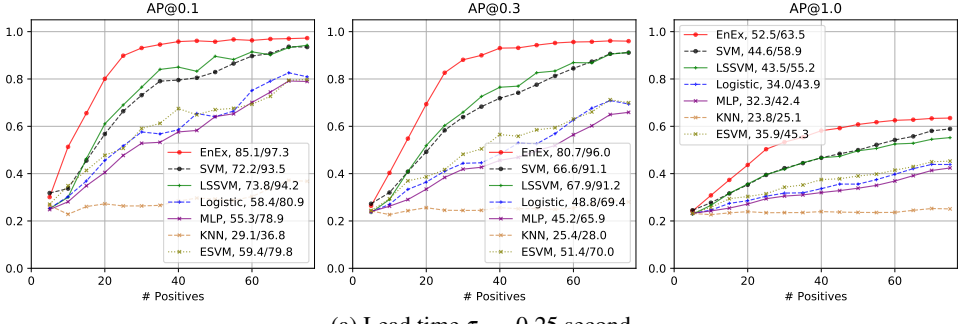


(a) $\tau_l = 2$s (b) $\tau_l = 4$s

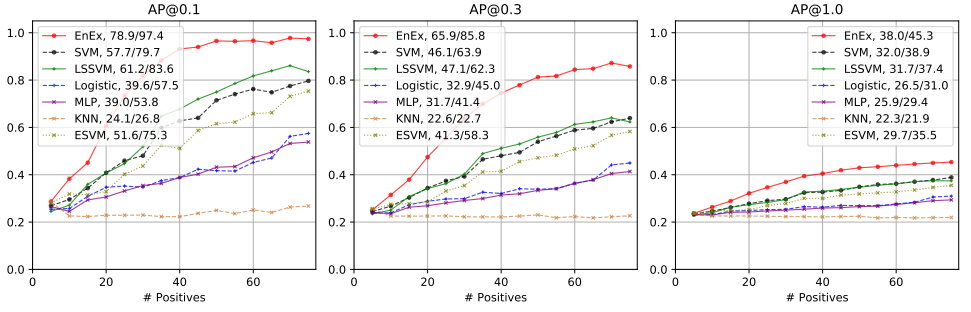
Figure 5: Average and final performance of different models under different recall thresholds. Experiments are run 10 times for predicting OpenDoor with lead times $\tau_l = 2$s and $\tau_l = 4$s.

8 Different lead times on PickupPhone action

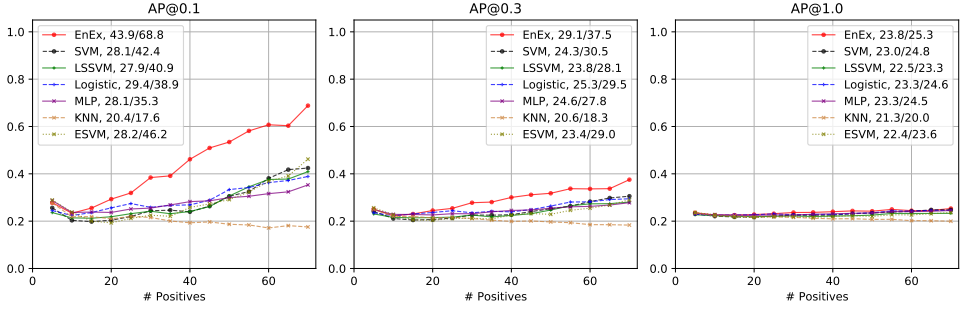
We use different lead times to predict the PickupPhone actions. The results are shown in Figure 6, which is an extension to Figure 3 in the paper. Note that the scores of the final models are also shown in the legends. And due to randomness in the experiments, the numbers are slightly different from the precision values in the paper.



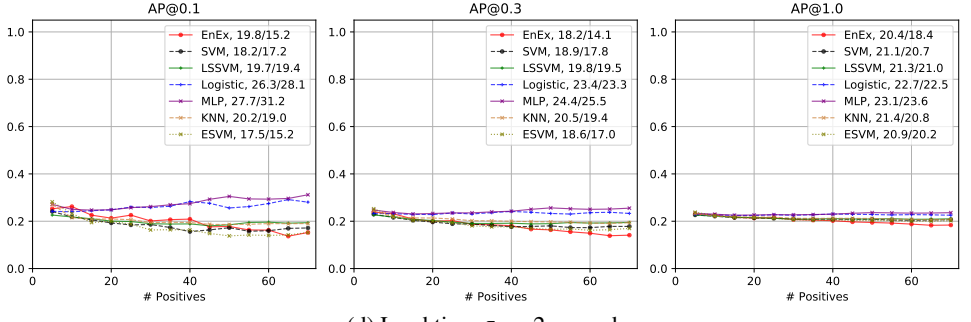
(a) Lead time $\tau_l = 0.25$ second



(b) Lead time $\tau_1 = 0.5$ second



(c) Lead time $\tau_1 = 1$ second



(d) Lead time $\tau_1 = 2$ second

Figure 6: Performance of different classifiers for predicting PickupPhone actions with different lead times. Each subplot shows the Average Precision at a particular recall threshold r (AP@r). Each curve in each subplot shows how the performance of a classifier changes as it encounters more target events and uses them for training. Each curve is the average of 40 experiment runs. The number next to each model in the legend box takes the form of average score/final score.