Supplementary Materials: Learning to Segment Actions from Visual and Language Instructions via Differentiable Weak Sequence Alignment

Yuhan Shen
Northeastern University
shen.yuh@northeastern.edu

Lu Wang
University of Michigan
wangluxy@umich.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

1. More Details on DWSA

1.1. Finding Alignment Path via Backtracking

Although we only need to compute the total alignment cost in our experiments, we can also easily obtain the optimal alignment path by backtracking. After updating the cumulative cost matrix $\boldsymbol{D}$, we can obtain the optimal alignment path via Algorithm 1.

Algorithm 1: Alignment Path Backtracking

```
input : Cumulative cost matrix $\boldsymbol{D}$
1 Initiate alignment path $\mathcal{P} = \varnothing$
2 $j \leftarrow 2q' + 1$
3 for $i \leftarrow q$ to $1$ do
4     if $j$ is odd then
5         $j \leftarrow \arg\min\{d_{i,1}, \cdots, d_{i,j}\}$
6     else
7         $j \leftarrow \arg\min\{d_{i,1}, \cdots, d_{i,j-1}\}$
8     add $(i, j)$ to $\mathcal{P}$
output: Alignment path $\mathcal{P}$
```

1.2. Backward Propagation

To obtain $\nabla_{\mathcal{O}}\text{DWSA}(\mathcal{O}, \mathcal{O}')$, we first calculate the derivatives of DWSA Loss $\mathcal{L}$ w.r.t. the entries of the cost matrix $\boldsymbol{\Delta}(\mathcal{O}, \mathcal{O}')$, which can be computed by,

$$\frac{\partial \mathcal{L}}{\partial \delta_{i,j}} = \frac{\partial \mathcal{L}}{\partial d_{i,j}} \cdot \frac{\partial d_{i,j}}{\partial \delta_{i,j}} = \frac{\partial \mathcal{L}}{\partial d_{i,j}}. \tag{1}$$

Let us denote the derivatives of the loss $\mathcal{L}$ w.r.t. $\delta_{i,j}$ as $g_{i,j}$, then

$$g_{i,j} = \frac{\partial \mathcal{L}}{\partial \delta_{i,j}} = \frac{\partial \mathcal{L}}{\partial d_{i,j}}. \tag{2}$$

Recall that the DWSA loss function, $\mathcal{L}$, is obtained by

$$\mathcal{L} = \min\nolimits_{\beta}\{d_{q,1}, d_{q,2}, \ldots, d_{q,2q'+1}\}. \tag{3}$$

Algorithm 2: Backward Propagation for DWSA

```
input : Matching cost $\boldsymbol{\Delta} \in \mathbb{R}^{q \times 2q'+1}$; Cumulative cost
        $\boldsymbol{D}$; soft-min parameter $\beta \geq 0$
1 $g_{q,j} \leftarrow \frac{e^{-d_{q,j}/\beta}}{\sum_{r=1}^{2q'+1} e^{-d_{q,r}/\beta}}, j \in \{1, \ldots, 2q'+1\}$
2 for $i \leftarrow q-1$ to $1$ do
3     for $j \leftarrow 2q'+1$ to $1$ do
4         if $j$ is odd then
5             $g_{i,j} \leftarrow \sum_{r \geq j} g_{i+1,r} e^{(-d_{i,j} + d_{i+1,r} - \delta_{i+1,r})/\beta}$
6         else
7             $g_{i,j} \leftarrow \sum_{r > j} g_{i+1,r} e^{(-d_{i,j} + d_{i+1,r} - \delta_{i+1,r})/\beta}$
8 Set $\boldsymbol{G} = [g_{i,j}]$
output: $\nabla_{\mathcal{O}}\text{DWSA}(\mathcal{O}, \mathcal{O}') = \big(\frac{\partial \boldsymbol{\Delta}(\mathcal{O}, \mathcal{O}')}{\partial \mathcal{O}}\big)^T \boldsymbol{G}$
```

Calculating the derivative of $\mathcal{L}$ w.r.t. $d_{q,j}$, we have

$$g_{q,j} = \frac{\partial \mathcal{L}}{\partial d_{q,j}} = \frac{e^{-\frac{d_{q,j}}{\beta}}}{\sum_{r=1}^{2q'+1} e^{-\frac{d_{q,r}}{\beta}}}. \tag{4}$$

Considering the formula of dynamically updating $\boldsymbol{D}$, for $j \leq 2n - 1$,

$$\frac{\partial d_{i+1,2n-1}}{\partial d_{i,j}} = \frac{e^{-\frac{d_{i,j}}{\beta}}}{\sum_{r=1}^{2n-1} e^{-\frac{d_{i,r}}{\beta}}}, \tag{5}$$

$$\begin{aligned} \beta \log \frac{\partial d_{i+1,2n-1}}{\partial d_{i,j}} &= -d_{i,j} - \beta \log \sum_{r=1}^{2n-1} e^{-\frac{d_{i,r}}{\beta}} \\ &= -d_{i,j} + \min_{\beta}\{d_{i,1}, \cdots, d_{i,2n-1}\} \\ &= -d_{i,j} + d_{i+1,2n-1} - \delta_{i+1,2n-1}. \end{aligned} \tag{6}$$

So, we obtain

$$\frac{\partial d_{i+1,2n-1}}{\partial d_{i,j}} = \exp\{\frac{-d_{i,j} + d_{i+1,2n-1} - \delta_{i+1,2n-1}}{\beta}\}. \tag{7}$$

	ProceL				CrossTask			
	K=7	K=10	K=12	K=15	K=7	K=10	K=12	K=15
Alayrac et al. [1]	3.67	4.90	5.19	5.54	3.39	4.31	4.42	4.46
Kukleva et al. [4]	18.98	16.68	16.46	14.99	15.67	13.87	11.92	10.98
Elhamifar et al. [2]	14.00	12.40	12.80	11.80	16.20	16.20	16.20	16.30
kmeans (visual)	16.57±0.15	15.82±0.25	15.29±0.22	14.67±0.27	19.13±0.09	17.35±0.07	16.24±0.09	15.26±0.11
SOPL (visual)	**21.18**±0.14	20.17±0.24	20.02±0.22	19.01±0.24	20.20±0.06	18.64±0.10	17.46±0.24	16.04±0.22
SOPL+soft-DTW	20.25±0.12	20.18±0.26	19.02±0.22	18.48±0.18	19.75±0.09	18.27±0.07	17.02±0.12	16.08±0.24
SOPL+DWSA	21.13±0.20	**20.67**±0.28	**20.30**±0.23	**19.46**±0.15	**20.50**±0.12	**19.12**±0.14	**17.63**±0.08	**16.55**±0.17

Table 1: Average F1 score on ProceL and CrossTask for different procedure lengths, K.

Similarly, for $j \leq 2n-1$, we have

$$\frac{\partial d_{i+1,2n}}{\partial d_{i,j}} = \exp\{\frac{-d_{i,j} + d_{i+1,2n} - \delta_{i+1,2n}}{\beta}\}. \quad (8)$$

Combining the above two equations, we get

$$\frac{\partial d_{i+1,r}}{\partial d_{i,j}} = \exp\{\frac{-d_{i,j} + d_{i+1,r} - \delta_{i+1,r}}{\beta}\}, \quad (9)$$

which holds for $j \leq 2\lceil\frac{r}{2}\rceil - 1$ (i.e., $r \geq 2\lfloor\frac{j}{2}\rfloor + 1$), otherwise the derivative is 0. As a result, we have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial d_{i,j}} &= \sum_{k=1}^{2q'+1} \frac{\partial \mathcal{L}}{\partial d_{q,k}} \frac{\partial d_{q,k}}{\partial d_{i,j}} \\ &= \sum_{k=1}^{2q'+1} \frac{\partial \mathcal{L}}{\partial d_{q,k}} \cdot \sum_{r=2\lfloor\frac{j}{2}\rfloor+1}^{2q'+1} \frac{\partial d_{q,k}}{\partial d_{i+1,r}} \cdot \frac{\partial d_{i+1,r}}{\partial d_{i,j}} \\ &= \sum_{r=2\lfloor\frac{j}{2}\rfloor+1}^{2q'+1} \sum_{k=1}^{2q'+1} \frac{\partial \mathcal{L}}{\partial d_{q,k}} \cdot \frac{\partial d_{q,k}}{\partial d_{i+1,r}} \cdot \frac{\partial d_{i+1,r}}{\partial d_{i,j}} \\ &= \sum_{r=2\lfloor\frac{j}{2}\rfloor+1}^{2q'+1} \frac{\partial \mathcal{L}}{\partial d_{i+1,r}} \cdot \frac{\partial d_{i+1,r}}{\partial d_{i,j}};\end{aligned} \quad (10)$$

$$\begin{aligned} g_{i,j} &= \sum_{r=2\lfloor\frac{j}{2}\rfloor+1}^{2q'+1} g_{i+1,r} \cdot \frac{\partial d_{i+1,r}}{\partial d_{i,j}} \\ &= \sum_{r=2\lfloor\frac{j}{2}\rfloor+1}^{2q'+1} g_{i+1,r} \cdot \exp\{\frac{-d_{i,j} + d_{i+1,r} - \delta_{i+1,r}}{\beta}\} \\ &= \begin{cases} \sum_{r\geq j} g_{i+1,r} \cdot e^{\frac{-d_{i,j}+d_{i+1,r}-\delta_{i+1,r}}{\beta}}, & l \text{ is odd} \\ \sum_{r>j} g_{i+1,r} \cdot e^{\frac{-d_{i,j}+d_{i+1,r}-\delta_{i+1,r}}{\beta}}, & l \text{ is even.} \end{cases}\end{aligned} \quad (11)$$

Therefore, we have obtained a backward recursion to compute the entire gradient matrix $\boldsymbol{G} = [g_{i,j}]$. The backward propagation is summarized in Algorithm 2.

Task	kmeans	SOPL	gain	OCE	Repeat Freq.
tie tie	10.13	20.12	98.6	5.8	3.8
change battery	15.77	28.71	82.1	10.4	0.2
change tire	18.97	29.27	54.3	14.0	7.2
setup chromecast	20.86	26.67	27.9	16.0	0.9
change toilet seat	13.06	15.28	17.0	16.8	1.1
make coffee	13.13	15.05	14.6	22.7	12.1
jump car	7.65	10.43	36.3	24.4	2.4
make pbj sand.	16.85	17.01	0.9	25.3	24.6
perform cpr	17.85	16.66	-6.7	26.1	34.1
make salmon sand.	28.93	30.53	5.5	29.4	15.3
assemble clarinet	10.04	11.56	15.1	35.1	3.5
repot plant	15.39	19.47	26.5	38.0	34.2

Table 2: The relative gain (%) in F1 score (%) from introducing time-stamp prototypes on ProceL, along with the ordering consistency error (OCE, in %, smaller is more consistent) and frequency of repeated steps (%) of each task.

2. More Details on Experiments

2.1. Effect of the Number of Prototypes

To be consistent with a few prior works that do not use the actual number of key-steps, we also set the number of prototypes/clusters K to be a predefined value, and report the performance with respect to different values of $K \in \{7, 10, 12, 15\}$ in Table 1. Notice that similar to the results presented in the main paper, our proposed method significantly improves the performance compared with unsupervised baselines in all settings across all values of K.

2.2. Effect of Time-Stamp Prototypes in SOPL

To better investigate the effect of introducing time-stamp prototypes in Soft Ordered Prototype Learning (SOPL), we compare the difference between kmeans and SOPL. Table 2 shows the relative gain in F1 score after adding time-stamp prototypes for each task. We also show the ordering consistency error and frequency of repeated key-steps for each task on ProceL (see below for formulation of these metrics).

We observe that using time-stamp prototypes generally helps more in tasks with more consistent ordering (e.g.,

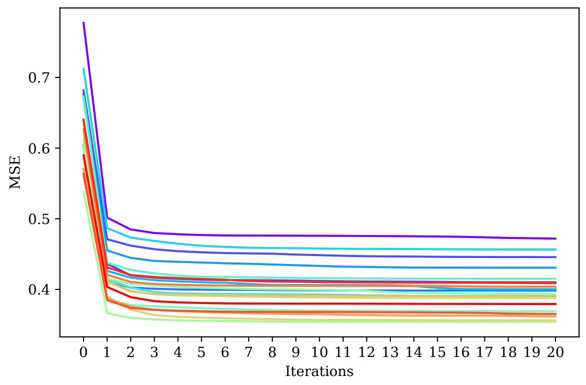


Figure 1: MSE as a function of the number of SOPL iterations, p for several randomly selected learning epochs.

'tie tie' or 'change battery'), and makes less or negative improvement for tasks with less consistent ordering (e.g., 'make salmon sandwich') or more repeated steps (e.g., 'perform CPR'). That explains why the improvement by SOPL is smaller on CrossTask, since the videos have higher average ordering consistency error on CrossTask compared to ProceL (27.7% v.s. 22.0%) and higher average frequency of repeated steps (32.2% v.s. 11.6%).

We compute ordering consistency error and frequency of repeated steps similar to [3, 1].

Order consistency error (OCE), denoted by O, is defined as 1 minus the average over the consistencies between distinct pairs of videos. The consistency between a pair of videos is defined as the ratio of the length of the longest common subsequence and the minimum number of annotated key-steps of two videos. This score is between 0 and 1 and lower error means higher order consistency between pairs of videos,

$$O \triangleq 1 - \sum_{i,j \in 1:N, i \neq j} \frac{l_{ij}}{c_{ij}}. \tag{12}$$

Frequency of repeated steps, denoted by R, is defined as 1 minus the ratio of unique key-steps and total number of annotated key-steps in videos. This score is between 0 and 1, and higher score implies more repetitions of key-steps across videos,

$$R \triangleq 1 - \frac{\sum_{n=1}^{N} u_n}{\sum_{n=1}^{N} g_n}. \tag{13}$$

2.3. Convergence of SOPL

In order to investigate the convergence of SOPL and the effect of the number of iterations, p, in SOPL, we set the maximum number of iterations to $p = 20$ and show the MSE as a function of the number of iterations, p, for several training epochs in Figure 1. As the figure shows, the MSE will typically converge after 3 to 5 iterations. In Table 3, we also report the average F1 score on ProceL and CrossTask when we set $p \in \{1, 3, 5, 7\}$. Notice that, generally, the difference between different iterations is small when $p \geq 3$, and the best performed choice is $p = 7$ on ProceL and $p = 5$ on CrossTask. We set p to be 5 for both datasets in all experiments.

	ProceL	Crosstask
p=1	20.05±0.11	20.46±0.08
p=3	20.72±0.16	20.87±0.09
p=5	21.07±0.25	**21.00**±0.09
p=7	**21.09**±0.10	20.96±0.06

Table 3: The average F1 score with respect to different numbers of iterations, p, in SOPL.

	ProceL	CrossTask
verb+dobj	20.41±0.21	20.27±0.04
verb-phrase	20.91±0.17	20.93±0.13
verb-phrase+conc. (ours)	**21.07**±0.25	**21.00**±0.09

Table 4: Average F1 on ProceL and CrossTask when using different methods of text processing.

2.4. Ablation Studies for Narration Processing

To investigate the impact of our text processing method, we conduct ablation studies on different ways of text processing. First, we experiment with *verb+dobj* pairs as in [1]. Then we study the effect of removing noisy verb-phrases based on concreteness scores, where we perform experiments on verb-phrases without removing phrases with low concreteness score. Table 4 shows the average F1 score on ProceL and CrossTask when using *verb+dobj* pairs and using verb-phrases without concreteness strategy. The F1 scores on verb-phrases are higher than *verb+dobj* pairs, because the verb-phrases can keep more necessary context. The performance without and with concreteness strategy are similar with the latter having slightly higher performance, which shows that our extra textual prototypes are able to model abstract phrases well.

2.5. Effects of Background Ratio

Figure 2 illustrates the effect of background ratio, b on performance of our method on CrossTask. We show the average F1, precision and recall with respect to different values of b. Notice that as b increases, the method will predict more frames as background, thus increasing precision and decreasing recall. As a result, F1 first increases and then goes down, and obtains the optimal value when $b = 0.4$. However, notice that for a large range of $b \in [0, 0.6]$, the F1 score is stable and only slightly changes.

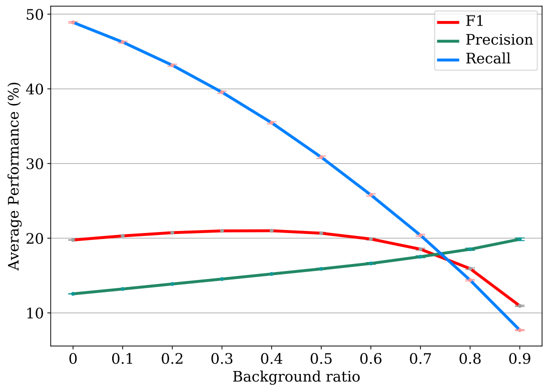


Figure 2: Average performance with respect to different values of background ratio, b, on CrossTask.

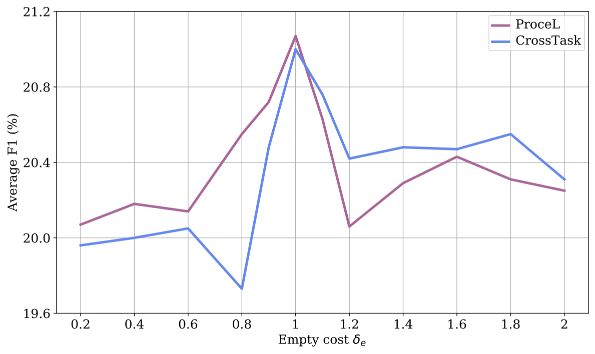


Figure 3: Average F1 with respect to different values of cost for empty alignment (δ_e).

2.6. Effect of Empty Alignment Cost

Figure 3 shows the effects of empty cost, δ_e, in DWSA. On both datasets, the method performs best when $\delta_e = 1$, yet, notice that the performances are close for all values of δ_e (for all values, our method outperforms the state of the art). When δ_e is small, the sequence alignment algorithm will align more rows with empty columns, so the model has less ability to learn informative features through the correspondence between two modalities. When δ_e is large, the algorithm disallows some visual prototypes to stay unmatched with the linguistic prototypes, thus, degrading the performance.

2.7. Qualitative Results on Key-Step Localization

We show more qualitative examples on key-step localization for three videos from three tasks in both CrossTask and ProceL datasets in Figure 4. Notice that, compared to other methods, our algorithm is able to more accurately localize key-steps in videos.

2.8. Visualization of Text-Video Alignments

In Figure 5 and 6, we visualize several verb-phrases and video frames that fall into linguistic and visual clusters

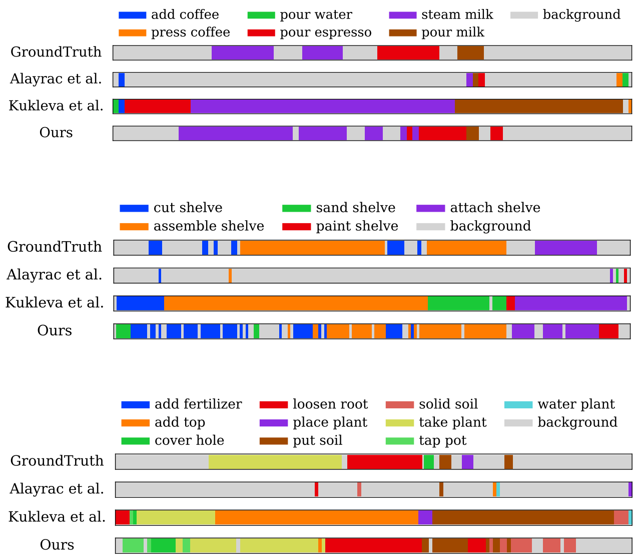


Figure 4: Localization results for three videos from the tasks 'make a latte' (top), 'build simple floating shelves' (middle), and 'repot plant' (bottom).

associated with each other via our DWSA algorithm. This allows us to visualize the correspondence between aligned prototypes of the two modalities.

For each pair of aligned linguistic and visual prototypes by DWSA, we randomly select 12 verb-phrases and 12 video frames, respectively. Due to the large quantity of background video segments, we limit the selection of video frames within the 50% video segments closest to visual prototypes. Notice that the verb-phrases that are assigned to the same linguistic prototype are highly similar in semantics. For example, in Figure 5, the verb-phrases in the first cluster are mostly related to 'remove screws', the ones in the second cluster are mostly related to 'pull battery', and the ones in the third cluster are mostly related to actions on 'screen'. In addition, most video frames are also depicting the actions mentioned in verb-phrases.

References

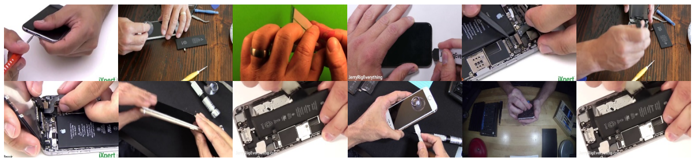
[1] J. B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 3

[2] E. Elhamifar and D. Huynh. Self-supervised multi-task procedure learning from instructional videos. *European Conference on Computer Vision*, 2020. 2

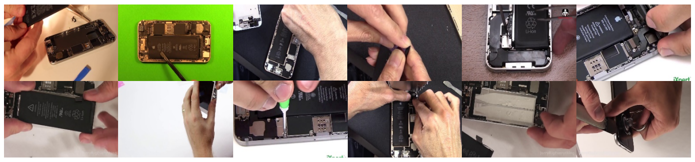
[3] E. Elhamifar and Z. Naing. Unsupervised procedure learning via joint dynamic summarization. *International Conference on Computer Vision*, 2019. 3

[4] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2

take the first two screws; put the screws;
take out the two screws; remove both screws;
remove these two screws; take out the bottom screws;
remove those screws; grab screws; undo these screws;
take off screw; have two screws; use screwdriver.



pull out the battery from the back; pull up the battery;
remove the battery in the future; reinstall your new battery;
reconnect the connecting cable for the battery;
feel a battery; reconnect the battery connector;
remove the battery from the rear case; pop battery in place;
grab my battery; lift the battery; recycle battery.



get the two screens in the bottom; press down the screen;
line up the screen; reassemble the LCD screen;
get your screen; close up the screen; crack your screen;
lower the screen; reach the home screen;
plug screen; hold your screen in place; clamp the screen.

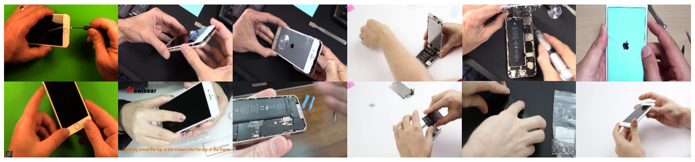
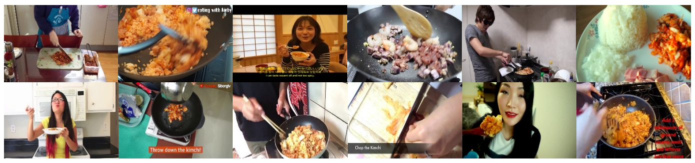


Figure 5: Randomly selected verb-phrases and video frames that fall into the same cluster, from the task ‘change iPhone battery’.

chop the kimchi; eat kimchi; cook out the kimchi;
add the kimchi after about a minute; use kimchi;
use Korean radish kimchi; mix rice with the radish kimchi;
have some kimchi; use whole cabbage kimchi;
wash kimchi; cook the fried kimchi; heat up the kimchi.



fry rice; love what with my fried rice; break up the rice;
add the rice into the pan; add the rice;
add the cold white rice; turn out rice; give this fried rice;
add the rice with no olive oil; fry rice at my list;
put sesame on the rice; make this for fried rice.

cook some food; check out the meat; cook up this;
cook meat for about 5 minutes; finish this cooking;
make this meal; cook that; cook the next thing;
have what in the refrigerator; turn the meat;
make dinner for your family; make food.

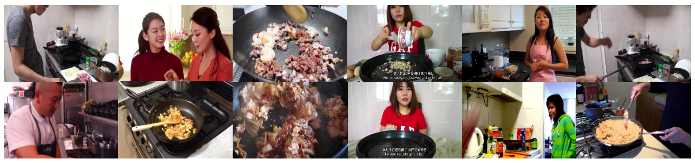


Figure 6: Randomly selected verb-phrases and video frames that fall into the same cluster, from the task ‘make kimchi fried rice’.