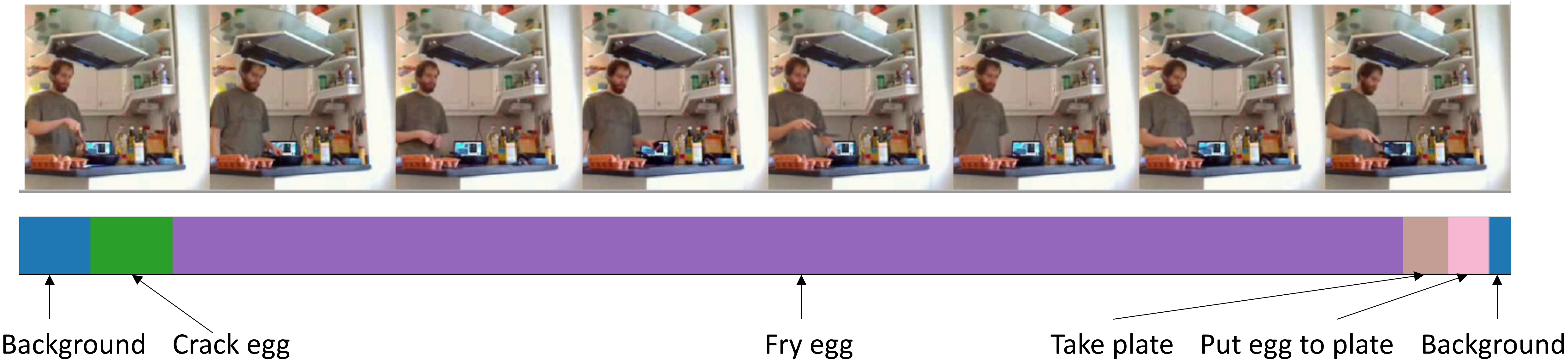
SSCAP: Self-supervised Co-occurrence Action Parsing for Unsupervised Temporal Action Segmentation

Zhe Wang, ***Hao Chen***, Xinyu Li, Chunhui Liu,
Yuanjun Xiong, Joseph Tighe, Charless Fowlkes

Amazon AWS AI

Introduction: Temporal Action Segmentation

Temporal action segmentation is a task to classify each frame in the video with an action label.



- Full frame level supervision is untrimmed video is expensive
- Weakly supervision reduces the cost, but still heavily relies on some non-trivial expertise in annotation

- Can we do it in an ***unsupervised*** manner?

Challenges: Unsupervised Action Segmentation

1. To extract highly distinguishable visual representations for each individual frame

2. To capture the temporal relations among frames and sub-actions, and thus to well estimate number and the order of the occurrence of each sub-action (i.e., the temporal path)

3. Even more challenging when dealing with videos that contain activities with complex structures and recurrence of sub-actions

SSCAP: Self-supervised Co-occurrence Action Parsing

1. To extract highly distinguishable visual representations for each individual frame

   SSCAP: uses self-supervised learning to extract features that are more temporal distinguishable

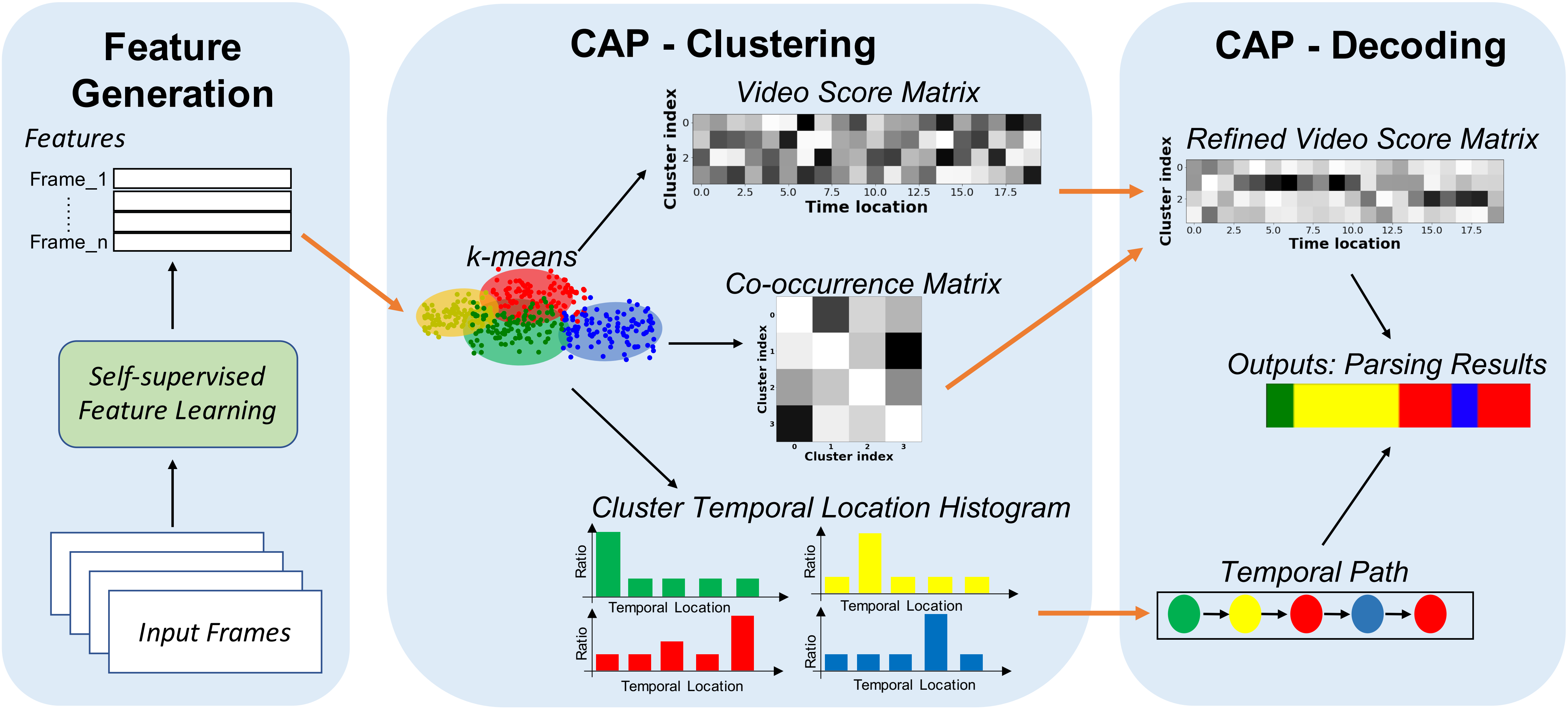
2. To capture the temporal relations among frames and sub-actions, and thus to well estimate number and the order of the occurrence of each sub-action (i.e., the temporal path)

   SSCAP: designs Co-occurrence Action Parsing (CAP) algorithm to estimate the temporal path and decode the frames into sub-actions, by

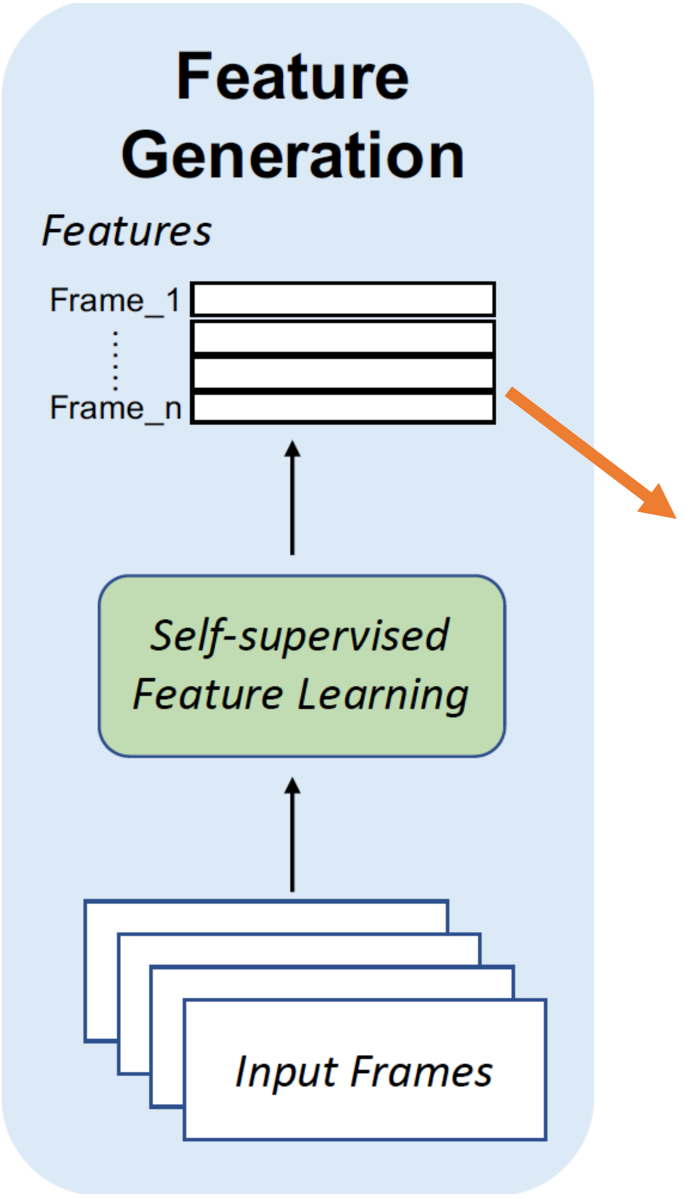
   - ***leveraging the estimated prior of the co-occurrence relations of sub-actions***
   - ***taking the recurrence of sub-actions into account and building the temporal location histogram***

SSCAP achieves SOTA result on Breakfast, Salad, and FineGym (with more complexed action structures), even outperforms weakly-supervised solutions.

SSCAP: Overview



SSCAP: Feature Generation



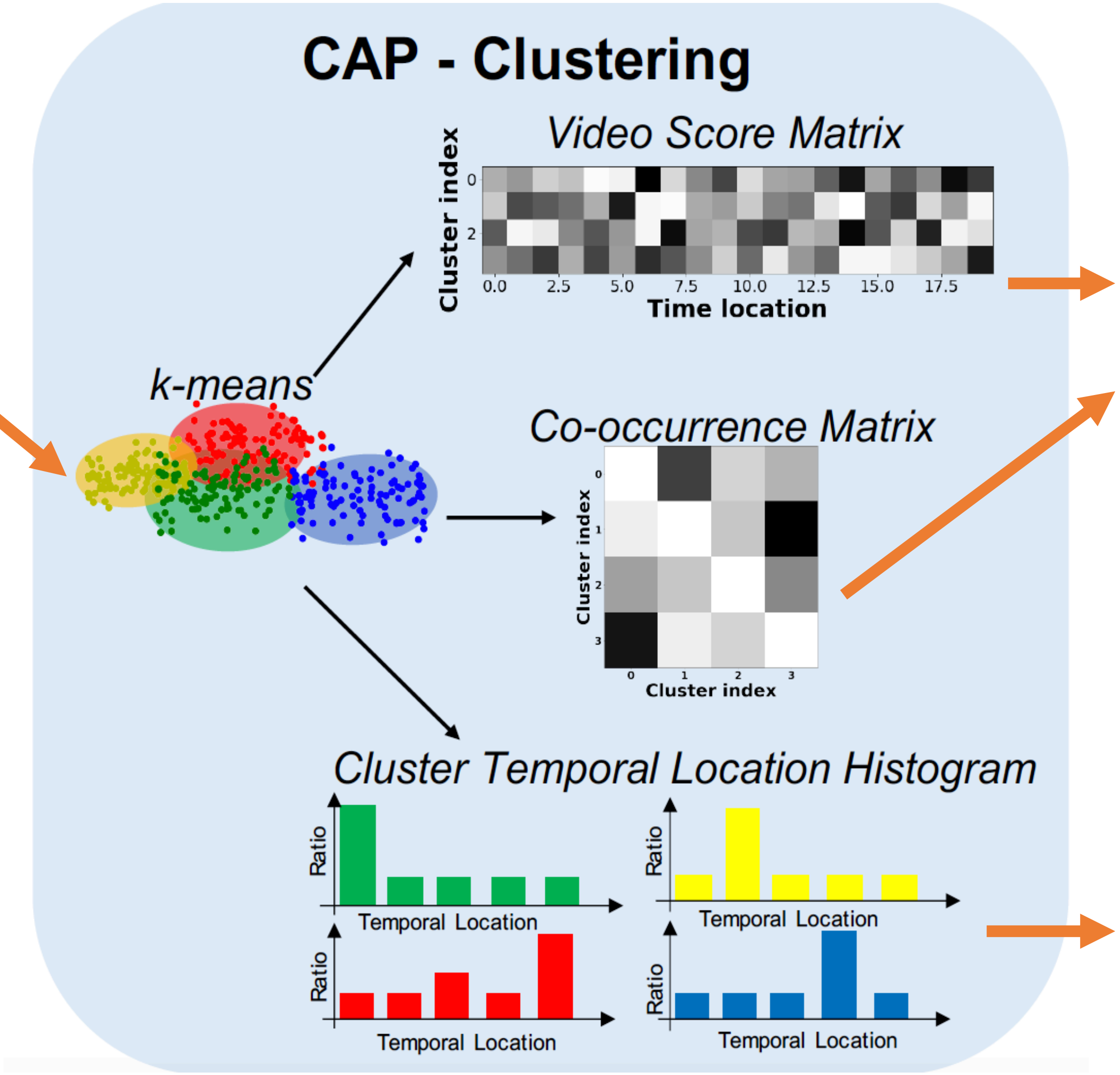
- **SpeedNet[1,2]:**
  - Pre-define four frame rate settings [2,4,8,15];
  - Randomly select one out of these four settings to generate clips;
  - Predict which frame-rate the clip is sampled from (i.e., a 4-way classification).

- **ShuffleLearn[3]:**
  - Shuffle M frames ($0 < M < N$) in a random order;
  - Flip the coin to decide whether to shuffle it or not when generating clips;
  - Predict whether a clip is shuffled or not.

- **RotationNet[4]:**
  - Pre-define four rotation degree settings [0,90,180,270], with some randomness when we rotate, i.e., in [−30,30] degree;
  - Randomly select one out of these four settings to rotate the whole clip;
  - Predict which of the settings is the rotation on (i.e., a 4-way classification).

[1] Dave Epstein et. al. “Oops! predicting unintentional action in video”. In CVPR, 2020.
[2] Sagie Benaim et. al. “Speednet: Learning the speediness in videos”. In CVPR, 2020.
[3] Ishan Misra et. al. "Shuffle and learn: Unsupervised learning using temporal order verification”. In ECCV, 2016.
[4] Spyros Gidaris et. al. “Un-supervised representation learning by predicting image rotations”. In ICLR, 2018.

SSCAP: CAP - Clustering



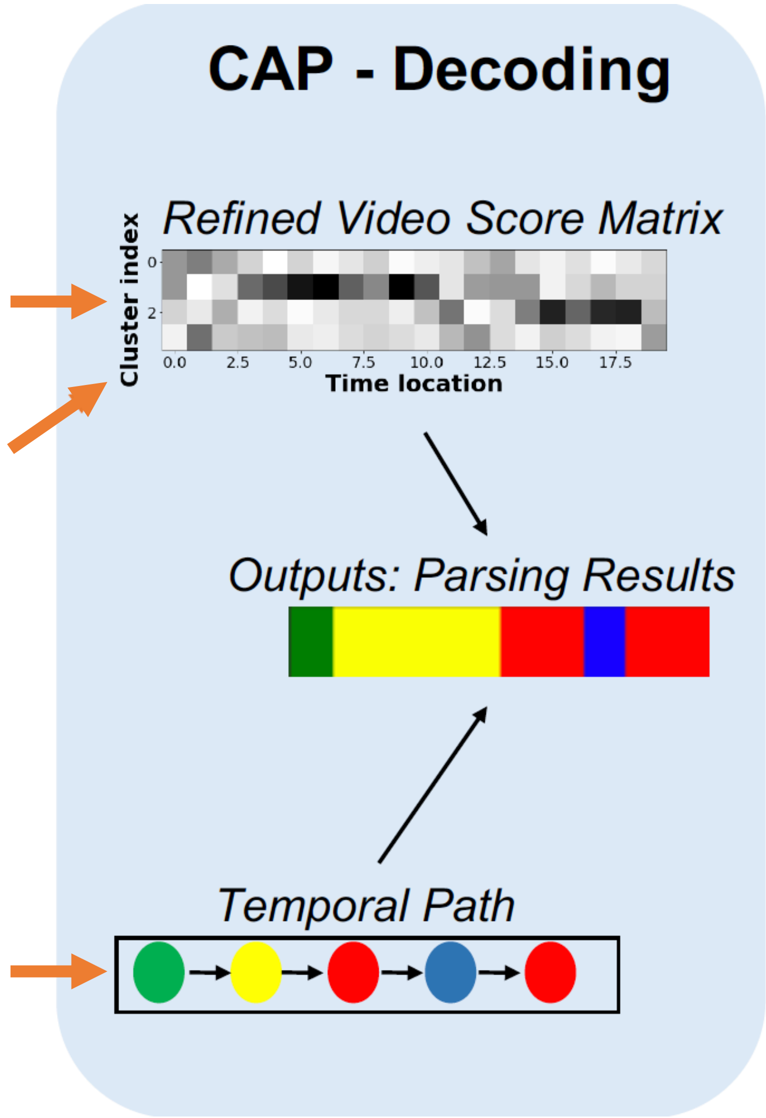
- ***Clustering:***
  - We extract all the frame-level features and cluster them into K-clusters using k-means (assuming K different sub-actions);

- ***Video Score Matrix $\boldsymbol{S} \in \mathbb{R}^{K \times N}$:***
  - Capture the score of each frame belonging to a cluster
  - $s_{(n,k)} = p(x_n|k) = \mathcal{N}(x_n; \mu_k, \Sigma_k)$

- ***Co-occurrence Matrix $\boldsymbol{C} \in \mathbb{R}^{K \times K}$:***
  - Capture the correlations among sub-actions underlying the native structure of activities
  - $c_{(i,j)} = \frac{O(i,j)}{O(j)}$, a conditional co-occurrence probability

- ***Cluster Temporal Location Histogram** $H(t_n, k)$**:***
  - Estimate where each cluster generally locates in temporal dimension
  - $t_n = \frac{n}{N}$, the relevant timestamp in the video
  - For action recurrence each cluster may have multiple significant bins in the histogram

SSCAP: CAP - Decoding



- **Refined Video Score Matrix $R \in \mathbb{R}^{K \times N}$:**
  - Capture correlation information among sub-actions and global patterns of the activity structures;
  - Carefully select clusters k to avoid over-segmenting the video to the non-existing classes.

Refined Video Score Matrix

- Initialization: $\mathcal{G} \leftarrow k_0$ (k_0 is the cluster with the largest ratio of frames $r(k_0)$ in current video).

- $k^* = k_0$

while $len(\mathcal{G}) \leq K$ *and* $r(k^*)>0$ **do**

1. For each remaining cluster $j \notin \mathcal{G}$:
   - update the video score matrix conditioned on the previous selected cluster k^*:
     $R_m[j, n] = P(j|k^*) \cdot S_m[j, n]$
2. Select the next cluster: $k^* \leftarrow \arg\max_j r(j)$.
3. Update: $\mathcal{G} \leftarrow \mathcal{G} \cup \{k^*\}$

end

- **Temporal Path Estimation and Decoding:**
  - Capture the multi-occur sub-actions and bi-directional sub-action transition;
  - For each cluster, we select top-K bins from temporal location histogram;
  - Then we concatenate the selected bins from all the clusters and order them into a time sequence based on their temporal locations;
  - Decoding: Viterbi algorithm[1]

[1] T. Quach and M Farooq. Maximum likelihood track formation with the Viterbi algorithm. In IEEE Conference on Decision and Control, 1994.

Results: Comparing with SOTA

Breakfast	MoF	F1 score
Unsupervised setting		
GMM [48]	0.346	-
LSTM + AL [1]	0.429*	-
CTE [30]	0.418	0.264
VTE-UNET [53]	0.481	-
ASAL [35]	0.525	0.379
Our SSCAP	0.511	**0.392**
Weakly-supervised setting		
Action Sets [45]	0.284	-
NNviterbi [46]	0.430	-
SCT [16]	0.304	-
SetViterbi [34]	0.408	-
EnergySeg [33]	0.630	-
Fully-supervised setting		
HTK [28]	0.259	-
GTRM [24]	0.650	-
MS-TCN [15]	0.663	-
BCN [60]	0.704	-

50Salads	MoF	F1 score
Unsupervised setting		
LSTM + AL [1]	0.606*	-
CTE [30]	0.355	-
VTE-UNET [53]	0.306	-
ASAL [35]	0.392	-
Our SSCAP	**0.414**	**0.303**
Weakly-supervised setting		
NNviterbi [46]	0.494	-
EnergySeg [33]	0.547	-
Fully-supervised setting		
HTK [28]	0.247	
GTRM [24]	0.826	-
MS-TCN [15]	0.734	-
BCN [60]	0.844	-

FineGym	MoF	F1 score
Baseline [30]	0.294	0.167
Our SSCAP	**0.666**	**0.297**

- *SSCAP achieves SOTA on both Breakfast and 50Salads datasets in the unsupervised setting;*
- *SSCAP on Breakfast even outperforms most of the weakly-supervised solutions;*
- *On FineGym (the challenging dataset), SSCAP achieves significant improvement to baseline, demonstrating the effectiveness of it in handling videos with more complex structures.*

Results: Ablation Studies

Dataset	SS	C-Matrix	M-T-Path	MoF	F1
Breakfast				0.418	0.264
	✓			0.508	0.391
	✓	✓		0.511	0.392
	✓	✓	✓	**0.511**	**0.392**
50Salads				0.355	-
	✓			0.372	0.281
	✓	✓		0.378	0.290
	✓	✓	✓	**0.414**	**0.303**
FineGym				0.294	0.167
	✓			0.425	0.246
	✓	✓		0.442	0.248
	✓	✓	✓	**0.666**	**0.297**

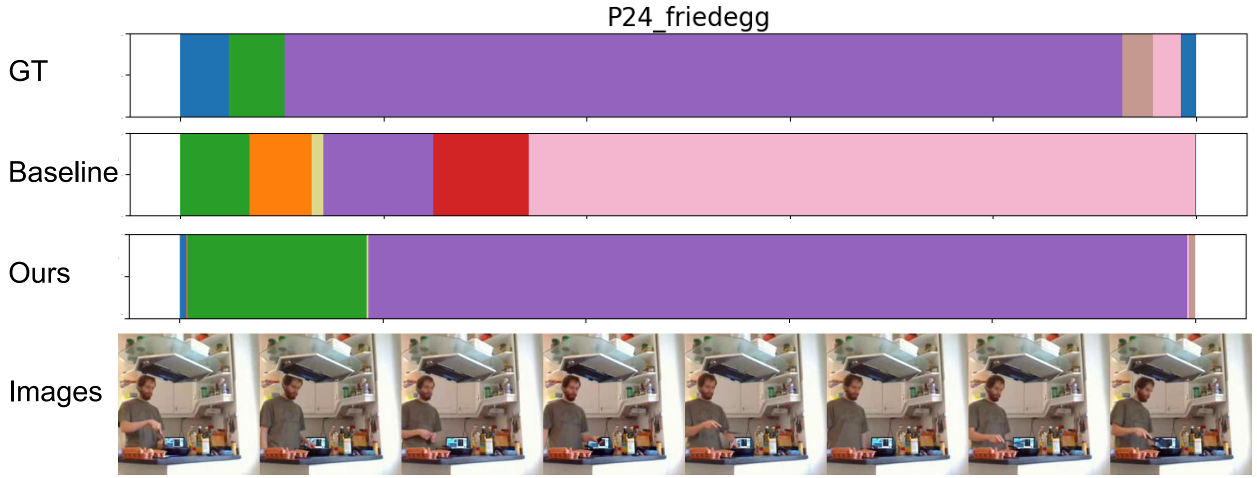
- ***Self-supervised learning always helps;***
- ***Co-occurrence matrix always helps, while on FineGym the improvement is more notable, indicating the importance of using the co-occurrence matrix while handling more complex scenarios;***
- ***Multi-occur temporal path helps 50Salads and FineGym, but not Breakfast, as most of the sub-actions only occur once in Breakfast. The improvement on FineGym is significant.***

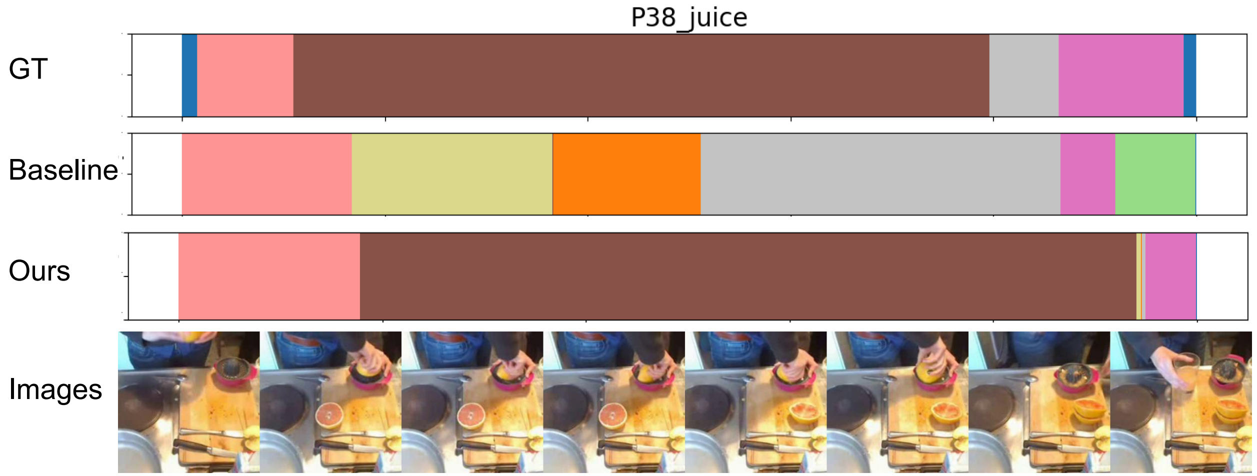
Results: Ablation Studies

	Feature Description	MoF	F1
Baseline			
(a)	IDT [30]	0.316	-
(b)	K400 I3D [30]	0.251	-
(c)	CTE [30]	0.418	0.264
Self-supervised on K400			
(d)	K400 SpeedNet	**0.508**	**0.391**
(e)	K400 RotationNet	0.328	0.317
(f)	K400 shuffleLearn	0.339	0.328
Self-supervised on Breakfast			
(g)	Breakfast SpeedNet	0.344	0.327
(h)	Breakfast RotationNet	0.307	0.319
(i)	Breakfast shuffleLearn	0.315	0.309
Self-supervised first on K400, then on Breakfast			
(j)	K400, Breakfast, SpeedNet	0.501	0.337
(k)	K400, Breakfast, RotationNet	0.279	0.290
(l)	K400, Breakfast, shuffleLearn	0.292	0.318

- *Self-supervised features always perform better than classical I3D feature pre-trained on Kinetics, indicating it's efficiency;*
- *RotationNet consistently performs worse than SpeedNet and ShuffleLearn, indicating that self-supervised from temporal augmentation is important;*
- *SpeedNet, as one of the most emerging video self-supervised learning approaches, performs the best;*
- *Larger dataset like Kinetics can help build better self-supervised representation, while smaller ones contain less variety. It's not needed to use target dataset to get a good feature re-presentation for the temporal action segmentation task.*

Results: Visualization



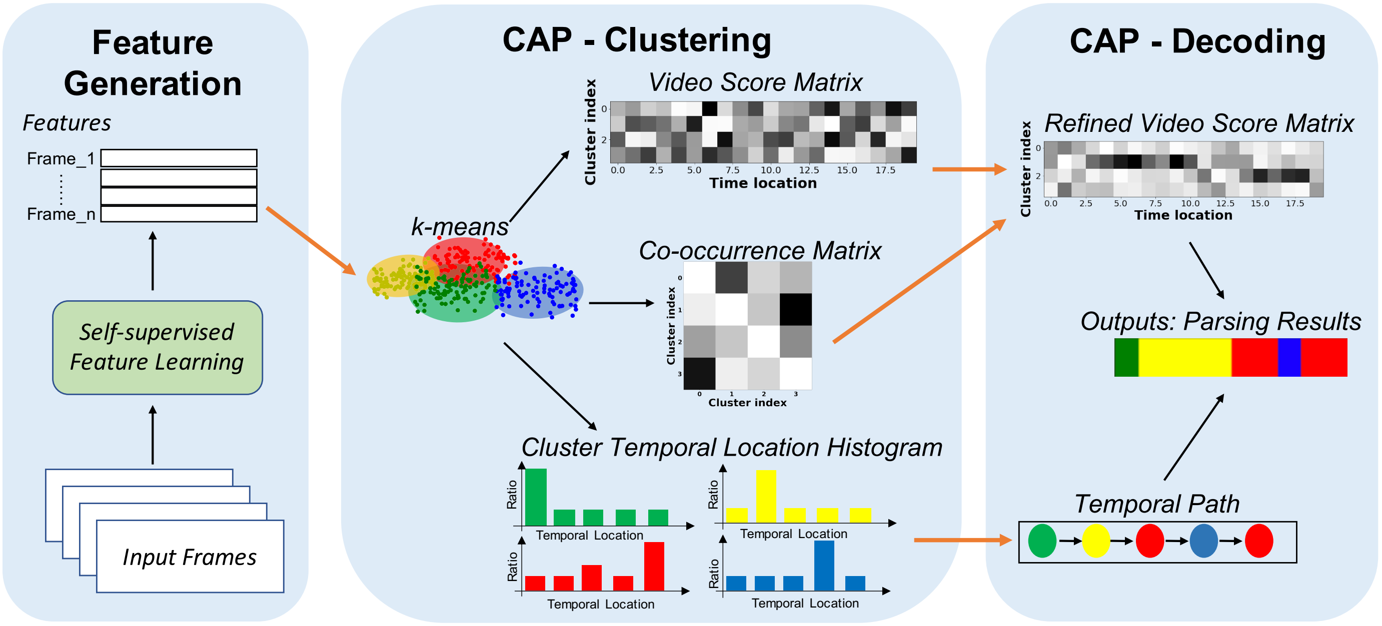


- *SSCAP is able to notably improve the segmentation quality;*

- *CAP algorithm is able to effectively suppressing the over-segmentation issue by introducing the co-occurrence relations among sub-actions in decoding;*

Conclusion

- Proposed SSCAP, an unsupervised temporal action segmentation solution that:
  - uses self-supervised methods in feature learning;
  - designs a co-occurrence action parsing algorithm that helps model the correlation among sub-actions and better handle complex activity structures in videos.
- SSCAP:
  - has achieved SOTA performance on three public benchmarks in unsupervised setting;
  - has even outperformed several recently proposed weakly-supervised methods;
  - is best designed from activities with complex action structures.



Contact:
Hao Chen: hxen@amazon.com
Zhe Wang: zwang15@uci.edu