Unsupervised Procedure Learning via Joint Dynamic Summarization

Ehsan Elhamifar
Khoury College of Computer Sciences
Northeastern University
eelhami@ccs.neu.edu

Zwe Naing
Khoury College of Computer Sciences
Northeastern University
naing.z@husky.neu.edu

Abstract

We address the problem of unsupervised procedure learning from unconstrained instructional videos. Our goal is to produce a summary of the procedure key-steps and their ordering needed to perform a given task, as well as localization of the key-steps in videos. We develop a collaborative sequential subset selection framework, where we build a dynamic model on videos by learning states and transitions between them, where states correspond to different subactivities, including background and procedure steps. To extract procedure key-steps, we develop an optimization framework that finds a sequence of a small number of states that well represents all videos and is compatible with the state transition model. Given that our proposed optimization is non-convex and NP-hard, we develop a fast greedy algorithm whose complexity is linear in the length of the videos and the number of states of the dynamic model, hence, scales to large datasets. Under appropriate conditions on the transition model, our proposed formulation is approximately submodular, hence, comes with performance guarantees. We also present ProceL, a new multimodal dataset of 47.3 hours of videos and their transcripts from diverse tasks, for procedure learning evaluation. By extensive experiments, we show that our framework significantly improves the state of the art performance.

1. Introduction

There exists a large amount of instructional data on the web in different forms. YouTube has over 500 million video search results for the phrase 'how to', with more than 400,000 results for tasks such as 'how to assemble a bike' or 'how to setup chromecast'. Such instructional data provide rich information for procedure[1] learning, which is to automatically learn the sequence of key-steps to perform a certain task. Procedure learning can be used to design autonomous agents that can perform complex tasks [1], to build knowledge bases of instructions, or to generate succinct procedures when human cannot spend time synthesizing the information from multiple sources [2]. Understanding instructional data at the scale necessary to build knowledge bases of thousands of tasks or to build assistive robots that respond to a large number of instructions, requires unsupervised procedure learning that does not rely on annotated data, which is complex and costly to gather.

Figure 1: The goal of unsupervised procedure learning is to learn and localize in data the sequence of key-steps to achieve a task. Ground-truth annotations and automatic discovery and localization of key steps via our method for the two tasks of 'perform CPR' (top) and 'replace iPhone battery' (bottom) for three videos in our new ProceL dataset are shown. We also show a few frames from each localized key-step found by our method.

Unsupervised procedure learning is an extremely challenging problem that needs to not only discover and localize the key-steps of a task, but also discover the logical ordering of the key-steps. This requires reconciling variations among instructions of a task, such as additional or missing steps and substantial amount of background actions, which are not related to the task. Over the past several years, we have seen interesting advances on the problem [3, 4, 5, 6, 7, 8].

[1] According to Webster's Encyclopedic Unabridged Dictionary of the English Language, 'Procedure' is defined as "the sequence of actions or instructions to be followed in solving a problem or completing a task."

The majority of existing work have focused on understanding procedures from narration [3, 4, 7, 9]. However, reliably obtaining text from spoken natural language using videos on the Internet is still a challenging problem, often requiring manual cleaning the results of automatic speech recognition systems. Moreover, to learn visual models of key-steps, existing methods assume that the text and visual information are aligned [4, 7, 9], which could be violated in real videos, e.g., human narrators first speak about one or multiple key-steps and then perform the subactions. Thus, to learn good visual models of key-steps, it is necessary to directly use visual data. Existing methods can handle and localize only one occurrence of a key-step in a video [5, 4, 9]. However, a procedure may contain repetitive key-steps, e.g., to perform 'CPR', one needs to alternate multiple times between 'give breath' and 'give compression' key-steps.

Paper Contributions. We address the problem of unsupervised procedure learning using visual data from unconstrained videos. We develop a joint dynamic summarization framework that produces a summary of the key-steps and their ordering and localizes the key-steps in videos. More specifically, given videos of the same task, we learn an Hidden Markov Model (HMM) whose latent states correspond to different subactivities, including background and key-steps. We develop an optimization that finds a subset of states that well represents all input videos jointly, while the sequence of states representing each video is compatible with the state transition model, hence, capturing the key-steps and their ordering.

Given that our proposed optimization is non-convex and NP-hard, we develop a fast greedy algorithm that incrementally grows the set of representatives by using dynamic programming at each iteration. The complexity of our algorithm is linear in the length of the videos and the number of states of the dynamic model, hence, scales to large number of long videos. Under appropriate conditions on the state transition model, our formulation is approximately submodular, hence it has performance guarantees. Our framework allows for repetitive key-steps in procedures and handles background subactivities in videos. By experiments, we show that our method significantly improves the state of the art, demonstrating the effectiveness of clustering-based summarization for procedure learning and localization.

Finally, we present a new multimodal dataset for procedure learning evaluation, consisting of 47.3 hours of videos from 12 diverse tasks with around 60 videos per task along with annotation of key-steps and manually cleaned captions in all videos. The tasks represent multiple domains, some with fine-grained detailed key-steps, e.g., 'replace iPhone battery' or 'assemble clarinet', and some containing interaction with virtual environments as in 'set up Chromecast'.

2. Related Work

Procedure Learning. The most related work to ours are [3, 4, 5]. Unlike our work, the goal of [3] is to segment individual videos rather than to produce a procedure summary for a task. [4] recovers a set of key-steps for each task, but it requires both visual and transcribed narration, where key-steps are discovered using information from the transcripts of videos and then are localized in videos. In contrast, our method uses only visual data, while it can be run on text as well. [5] develops an unsupervised iterative discriminative-generative method for segmentation of visual data into multiple sub-activities. However, it handles only one occurrence of a key-step in each video. Our method handles repeated key-steps and allows for missing or additional key-steps in videos.

Several recent work have addressed tasks related to procedure learning. Given an ordered or unordered list of subactions in videos, [10, 11, 12, 13] have studied the problem of assigning an action label to each frame in a video. However, this requires knowing the grammar or the dictionary of the task. In [14, 15], structured recipe texts are used to learn an action graph that represents the interaction between ingredients and actions being performed on them. However, the steps and ingredients are assumed to be given and known and they produce a relationship graph rather than a summary of the task. [7] focuses on aligning procedure steps in a video with written recipes, where steps are known in advance. This is different than our setting, where we want to discover key-steps. The work in [16] focuses on segmentation of procedure steps, but relies on supervised learning and does not produce a sequence of key-steps.

Subset Selection and Summarization. As we select representative states jointly among all videos, our work is related to collaborative summarization [17, 18]. Both these work aim to summarize one video using information from a collection of videos of the same theme. In contrast, our framework generates a common summary for all videos of a task and, more importantly, incorporates the sequential structure of data, necessary for discovering the ordering of key-steps.

The majority of existing subset selection methods [19, 20, 21, 22, 23, 24, 25, 26] address single video summarization and many cannot extract a common sequence of key-steps across videos of the same task. Moreover, they do not incorporate the dynamic model of key-steps across videos, often promoting sequential diversity that leads to selecting background subactions. While [27, 28] incorporate dynamic model, they works for single video summarization, where [27] relies on a computationally complex message passing algorithm, having quadratic and cubic complexity, respectively, in the video length and the number of states. Moreover, it is semi-supervised, using ground-truth annotations of a subset of videos to produce the final summary.

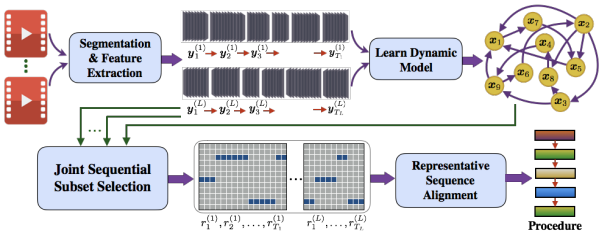


Figure 2: We develop an unsupervised procedure learning framework. Given videos of the same task, we segment videos, extract features from each segment and learn an HMM for the videos. We then find a joint sequential summarization of the input videos using the HMM, via a fast greedy method. Finally, we align sequences of representatives across videos to produce the procedure key-steps.

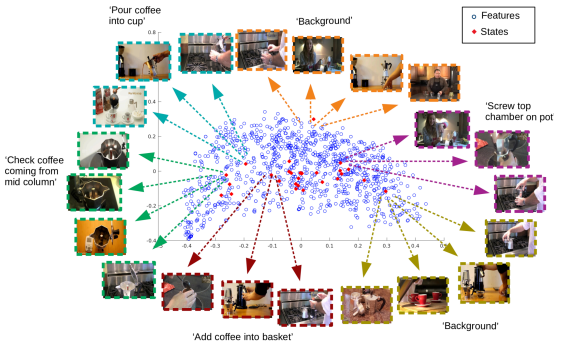


Figure 3: Visualization of segments and recovered hidden states of the HMM learned from videos of the task 'make coffee with moka pot' (for clarity, transition arrows are not shown). While some states correspond to key-steps, some correspond to unrelated background subactivities. We recover the key-states using our joint sequential subset selection framework.

3. Unsupervised Procedure Learning

Assume we have a collection of L instructional videos, $\mathcal{Y}_1, \ldots, \mathcal{Y}_L$, of the same task, from which we want to learn a concise procedure, i.e., key-steps and their ordering to achieve the task. Despite variations in videos, such as visual appearance, view points and length of videos as well as unrelated background subactions in every video, one can identify key-segments, where the underlying action in each segment is seen in many videos, as well as an ordering in the sequence of key-segments, common across most videos, see Figure 1. Our goal is to recover the common sequence of key-steps as the procedure description and localize the key-steps in all videos. To achieve this, we propose a framework that consists of the following components (see Figure 2).

- We segment each video and extract a feature vector from each segment, obtaining a time-series representation $\mathcal{Y}_\ell = (\boldsymbol{y}_1^{(\ell)}, \ldots, \boldsymbol{y}_{T_\ell}^{(\ell)})$ for each video ℓ. We then learn, from all input videos, an HMM, $(\mathcal{X}, \pi_0, \pi, p)$, where $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ corresponds to the set of distinct hidden states, π_0 is the initial probability, π is the transition probability between hidden states, i.e., $\pi(\boldsymbol{x}_{i'}|\boldsymbol{x}_i)$ for all i and i', and p denotes the observation emission probability from each state, i.e., $p(\boldsymbol{y}|\boldsymbol{x}_i)$ for all i. The set of hidden states $\mathcal{X}$ correspond to different subactivities across videos, including background and key-steps (see Figure 3), while π captures the ordering of the subactivities across videos and p denotes the likelihood of each segment belonging to a particular subactivity.
- Given that key-steps are common across many videos and their ordering must follow the transition probabilities, π, we develop a joint sequential subset selection optimization and a fast greedy maximization algorithm that selects a subset of hidden states from $\mathcal{X}$ that well encodes the input videos $\mathcal{Y}_1, \ldots, \mathcal{Y}_L$, where the sequence of representative hidden states for each video are compatible with the initial and transition probabilities.
- Our optimization recovers the same set of representative states for all videos, however, the sequences of assignments of segments to representatives could be different across videos. To create a single sequence of key-steps as the procedure description, we perform multiple sequence alignment [29, 4] on the sequences of representatives of videos. The solution of our optimization localizes key-steps as it finds the assignment of each video segment to each representative hidden state (i.e., each key-step).

In Section 5, we discuss the details of the first part on segmentation and feature extraction. Next, we describe the details of our optimization and the alignment approach.

3.1. Joint Sequential Subset Selection

Recall that, given L videos from the same task, $\mathcal{Y}_\ell = (\boldsymbol{y}_1^{(\ell)}, \ldots, \boldsymbol{y}_{T_\ell}^{(\ell)})$ denotes the time-series representation of the video ℓ with T_ℓ segments, where $\boldsymbol{y}_t^{(\ell)}$ is the feature vector of segment t of the video ℓ. As discussed above, we learn an HMM from the video time-series data to find the set of distinct hidden states, $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$, state initial probability, π_0, and state transition probabilities, π. To find the sequence of key-steps of the procedure and to localize each key-step in the videos, we propose a joint sequential subset selection framework that finds a representative subset of the hidden states, $\mathcal{S} \subseteq \mathcal{X}$, of size at most k, satisfying desired conditions described below, as well as the assignment of each video segment to each representative state.

Let $\mathcal{S}$ denote the unknown set of representatives from $\mathcal{X}$. With abuse of notation, we use $\mathcal{S}$ to refer to both the set of representative states and the set of indices of representative states. Let $r_t^{(\ell)} \in \mathcal{S}$ denote the index of the representative of $\boldsymbol{y}_t^{(\ell)}$, in other words, $\boldsymbol{y}_t^{(\ell)}$ is assigned to $\boldsymbol{x}_{r_t^{(\ell)}}$. We denote the assignment sequence for $\mathcal{Y}_\ell$ by $\boldsymbol{r}_\ell \triangleq (r_1^{(\ell)}, r_2^{(\ell)}, \ldots, r_{T_\ell}^{(\ell)}) \in \mathcal{S}^{T_\ell}$, where $\mathcal{S}^{T_\ell}$ is the cartesian product over $\mathcal{S}$.

Global Potential Function. The ideal summary must satisfy three properties: i) the size of the representative set, $|\mathcal{S}|$, must be small, ii) each assignment sequence $\boldsymbol{r}_\ell$ must well

encode $\mathcal{Y}_\ell$, iii) each sequence $\boldsymbol{r}_\ell$ must be compatible with the initial, π_0, and transition probabilities, π. To achieve these goals, we define a potential function $\Psi(\boldsymbol{r}_1,\dots,\boldsymbol{r}_L)$ and propose to solve

$$\max_{\mathcal{S}:\,|\mathcal{S}|\leq k}\ \max_{\{\boldsymbol{r}_\ell\in\mathcal{S}^{T_\ell}\}_{\ell=1}^{L}}\ \log\Psi(\boldsymbol{r}_1,\dots,\boldsymbol{r}_L), \tag{1}$$

over all sets $\mathcal{S}$ of size at most k and over all possible assignments $\boldsymbol{r}_\ell\in\mathcal{S}^{T_\ell}$. Given that the outer maximization in (1) restricts the size of the representative set, achieving the first goal, we define Ψ to reflect the two remaining goals of obtaining high encoding and dynamic compatibility for the assignment sequences. More specifically, we define

$$\Psi(\boldsymbol{r}_1,\dots,\boldsymbol{r}_L)\triangleq\Phi_{\text{enc}}(\boldsymbol{r}_1,\dots,\boldsymbol{r}_L)\times\Phi_{\text{dyn}}^{\beta}(\boldsymbol{r}_1,\dots,\boldsymbol{r}_L), \tag{2}$$

where $\Phi_{\text{enc}}(\boldsymbol{r}_1,\dots,\boldsymbol{r}_L)$ is an encoding potential that favors selecting a sequence of representative assignments $\boldsymbol{r}_\ell$ from $\mathcal{S}^{T_\ell}$ that well encodes $\mathcal{Y}_\ell$ for every ℓ and $\Phi_{\text{dyn}}(\boldsymbol{r}_1,\dots,\boldsymbol{r}_L)$ is a dynamic potential that favors sequences $\boldsymbol{r}_1,\dots,\boldsymbol{r}_L$ that are compatible with the state initial and transition probabilities. The regularization parameter $\beta\geq 0$ sets a trade-off between the two terms, where a small β results in discounting the dynamic compatibility of assignment sequences.

Encoding and Dynamic Potential Functions. Since the encoding of each segment of $\mathcal{Y}_\ell$ depends on its own representative, we consider a factorization of the encoding potential function as

$$\Phi_{\text{enc}}(\boldsymbol{r}_1,\dots,\boldsymbol{r}_L)=\prod_{\ell=1}^{L}\Big(\prod_{t=1}^{T_\ell}p\big(\boldsymbol{y}_t^{(\ell)}|\boldsymbol{x}_{r_t^{(\ell)}}\big)\Big)^{1/T_\ell}, \tag{3}$$

where $p(\boldsymbol{y}_t^{(\ell)}|\boldsymbol{x}_{r_t^{(\ell)}})$ is the likelihood that $\boldsymbol{y}_t^{(\ell)}$ is emitted from the hidden state $\boldsymbol{x}_{r_t^{(\ell)}}$. The exponent $1/T_\ell$ in (3) normalizes sequences of different lengths, so that each video contributes equally to the summarization, independent of its length. For notation simplicity, in the rest of the paper, we use $s_{r_t^{(\ell)},t}$ to denote the logarithm of emission probability,

$$s_{r_t^{(\ell)},t}\triangleq\log p\big(\boldsymbol{y}_t^{(\ell)}|\boldsymbol{x}_{r_t^{(\ell)}}\big). \tag{4}$$

On the other hand, we define the dynamic potential function to capture the compatibility of the sequences of the representatives, according to the learned state initial and transition probabilities, as

$$\Phi_{\text{dyn}}(\boldsymbol{r}_1,\dots,\boldsymbol{r}_L)=\prod_{\ell=1}^{L}\Big(\pi_0\big(r_1^{(\ell)}\big)\prod_{t=2}^{T_\ell}\pi\big(r_t^{(\ell)}|r_{t-1}^{(\ell)}\big)\Big)^{1/T_\ell}, \tag{5}$$

where, for a video ℓ, $\pi_0(i)$ denotes the probability of selecting $\boldsymbol{x}_i$ as the representative of $\boldsymbol{y}_1^{(\ell)}$ and $\pi(i'|i)$ denotes the probability of selecting $\boldsymbol{x}_{i'}$ as the representative of $\boldsymbol{y}_t^{(\ell)}$ given that $\boldsymbol{x}_i$ is the representative of $\boldsymbol{y}_{t-1}^{(\ell)}$.

To simplify the notation, we define

$$q_i^0\triangleq\log\pi_0(i),\quad q_{i,i'}\triangleq\log\pi(i'|i). \tag{6}$$

Using the definitions of the encoding and dynamic potentials in (3) and (5), we can write the logarithm of Ψ, which we want to maximize, as

$$\log\Psi=\sum_{\ell=1}^{L}\frac{1}{T_\ell}\Big(\sum_{t=1}^{T_\ell}s_{r_t^{(\ell)},t}+\beta\big(q^0_{r_1^{(\ell)}}+\sum_{t=2}^{T_\ell}q_{r_{t-1}^{(\ell)},r_t^{(\ell)}}\big)\Big). \tag{7}$$

Notice that when $\beta=0$ and $L=1$, the objective function above reduces to the well-known Facility Location (FL) function for summarization [30]. Next, we develop a fast algorithm, which runs linearly in the lengths of videos and number of states, to maximize (7).

3.2. Greedy Joint Sequential Subset Selection

In this section, we develop a fast greedy method to solve our optimization in (1) with the objective function given in (7). Notice that (1) consists of two maximizations: the outer maximization searches for the best subset of the states $\mathcal{S}$ of size at most k and the inner maximization searches for the best assignment sequences $\{\boldsymbol{r}_\ell\}_{\ell=1}^{L}$ using a fixed $\mathcal{S}$. Thus, we can rewrite the problem in (1) in the equivalent form

$$\max_{\mathcal{S}:|\mathcal{S}|\leq k}f(\mathcal{S}), \tag{8}$$

where the set function $f(\mathcal{S})$ (a set function is a function that assigns real values to sets) is defined as

$$\begin{aligned}f(\mathcal{S})&\triangleq\max_{\{\boldsymbol{r}_\ell\in\mathcal{S}^{T_\ell}\}_{\ell=1}^{L}}\log\Psi(\boldsymbol{r}_1,\dots,\boldsymbol{r}_L)\\&=\max_{\{\boldsymbol{r}_\ell\in\mathcal{S}^{T_\ell}\}_{\ell=1}^{L}}\sum_{\ell=1}^{L}\frac{1}{T_\ell}\sum_{t=2}^{T_\ell}w_{t-1,t}^{(\ell)}.\end{aligned} \tag{9}$$

Here, $\log\Psi$ is given by (7). However, for simplicity of notation and subsequent derivations, we have introduced the notation $w_{t-1,t}^{(\ell)}$ for the terms inside the first summation in (7). More specifically, for every ℓ in $\{1,\dots,L\}$, we have

$$w_{t-1,t}^{(\ell)}\triangleq\begin{cases}s_{r_{t-1}^{(\ell)},t-1}+\beta(q^0_{r_{t-1}^{(\ell)}}+q_{r_{t-1}^{(\ell)},r_t^{(\ell)}}), & \text{if } t=2,\\ s_{r_{t-1}^{(\ell)},t-1}+\beta q_{r_{t-1}^{(\ell)},r_t^{(\ell)}}, & \text{if } 2<t<T_\ell,\\ s_{r_{t-1}^{(\ell)},t-1}+s_{r_t^{(\ell)},t}+\beta q_{r_{t-1}^{(\ell)},r_t^{(\ell)}}, & \text{if } t=T_\ell.\end{cases} \tag{10}$$

The greedy algorithm [31, 30] for maximizing a set function $f(\cdot)$, starts with initializing an active set to the empty set, $\Lambda=\varnothing$, and incrementally grows the active set over k iterations. At each iteration, the greedy method adds to the current active set Λ the element i in $\{1,\dots,M\}\backslash\Lambda$ that achieves the highest value for $f(\Lambda\cup\{i\})$.

Computing $f(\Lambda)$ in (9), in principle, requires a combinatorial search over an exponentially large parameter space,

$\{\boldsymbol{r}_\ell \in \mathcal{S}^{T_\ell}\}_{\ell=1}^{L}$. However, we can use the sequential structure of videos to overcome this challenge and to efficiently perform evaluations of f. First, notice that we can write

$$\begin{aligned} f(\Lambda) &= \max_{\{\boldsymbol{r}_\ell \subseteq \Lambda^{T_\ell}\}_{\ell=1}^{L}} \sum_{\ell=1}^{L} \frac{1}{T_\ell} \sum_{t=2}^{T_\ell} w_{t-1,t}^{(\ell)} \\ &= \sum_{\ell=1}^{L} \frac{1}{T_\ell} \times \max_{\boldsymbol{r}_\ell \subseteq \Lambda^{T_\ell}} \Big(\sum_{t=2}^{T_\ell} w_{t-1,t}^{(\ell)} \Big), \end{aligned} \tag{11}$$

i.e., for a fixed set Λ, finding the optimal assignment sequence $\boldsymbol{r}_\ell \in \Lambda^{T_\ell}$ for each video can be done independently, hence, we perform L separate maximizations. We then use the fact that $\max_{\boldsymbol{r}_\ell \subseteq \Lambda^{T_\ell}} \sum_{t=2}^{T_\ell} w_{t-1,t}^{(\ell)}$ is a maximization over the sum of chains of variables, $r_1^{(\ell)}, r_2^{(\ell)}, \ldots, r_{T_\ell}^{(\ell)}$. Thus, we distribute the maximization over summation and compute

$$\begin{aligned} &f^{(\ell)}(\Lambda) \triangleq \max_{\boldsymbol{r}_\ell \subseteq \Lambda^{T_\ell}} \sum_{t=2}^{T_\ell} w_{t-1,t}^{(\ell)} \\ &= \max_{r_T^{(\ell)} \in \Lambda} \Big(w_{T-1,T}^{(\ell)} + \cdots + \max_{r_2^{(\ell)} \in \Lambda} \big(w_{2,3}^{(\ell)} + \max_{r_1^{(\ell)} \in \Lambda} w_{1,2}^{(\ell)} \big) \Big). \end{aligned} \tag{12}$$

As a result, we can exactly compute $f(\Lambda)$ (and similarly $f(\Lambda \cup \{i\})$) in each iteration of the greedy algorithm by using dynamic programming [32] in (12). Algorithm 1 shows the steps of our greedy algorithm.

Computational Complexity. As discussed in the supplementary materials, the running time of our greedy algorithm is $O(k^2 M \sum_{\ell=1}^{L} T_\ell)$. In other words, our algorithm runs linearly in the number of states and the lengths of videos, hence, scales to large datasets. This is a significant improvement over message passing [27], whose complexity is $O(M^2 \sum_{\ell=1}^{L} T_\ell^2 + M^3 \sum_{\ell=1}^{L} T_\ell)$. This could be better seen by recalling that in our case $k < M \ll T_\ell$, i.e., the number of states is often much smaller than the length of videos.

3.3. Summary Alignment

Once we solve the proposed optimization in (1), we obtain the optimal subset of states $\mathcal{S}$ of size k as well as the sequence of assignments of segments of each video to the representative states, i.e., $\boldsymbol{r}_\ell = (r_1^{(\ell)}, r_2^{(\ell)}, \ldots, r_{T_\ell}^{(\ell)})$ for $\ell = 1, \ldots, L$. Notice that in our method, each video can use a subset of $\mathcal{S}$ with the sequence of key-step assignments being slightly different across videos, hence, we can handle additional or missing key-steps and slightly different ordering of key-steps in videos. To create a single sequence of key-steps as the procedure description for all the videos, we perform alignment on the assignment sequences.

Given that each optimal assignment sequence $\boldsymbol{r}_\ell$ often contains many repetitions, we first remove the repetitions from each sequence, and denote the resulting sequence of distinct consecutive states by $\boldsymbol{u}_\ell$. For example, for $\boldsymbol{r}_\ell = (3, 3, 3, 12, 12, 9, 9, 9, 9, 3, 3)$, we obtain $\boldsymbol{u}_\ell = (3, 12, 9, 3)$,

Algorithm 1 : Greedy Joint Sequential Subset Selection

Input: f defined in (9); Budget k.

```
Initialize: Λ = ∅;
for j = 1, ..., k do
    for i ∈ {1, ..., M}\Λ do
        for ℓ = 1, ..., L do
            Find f^(ℓ)(Λ ∪ {i}) and r_ℓ via dynamic programing;
        end for
        Compute f(Λ ∪ {i}) = Σ_{ℓ=1}^{L} (1/T_ℓ) f^(ℓ)(Λ ∪ {i});
    end for
    Compute i* = argmax_{i∈{1,...,M}\Λ} f(Λ ∪ {i});
    if f(Λ ∪ {i*}) > f(Λ) then
        Update Λ ← Λ ∪ {i*};
    else
        Break;
    end if
end for
Set S* = Λ and r*_ℓ = r_ℓ for ℓ = 1, ..., L;
```

Output: Representative set $\mathcal{S}^*$ of size k, assignment sequences $\{\boldsymbol{r}_\ell^*\}_{\ell=1}^{L}$.

which shows that the sequence of the representative states for the ℓ-th video are $\boldsymbol{x}_3 \rightarrow \boldsymbol{x}_{12} \rightarrow \boldsymbol{x}_9 \rightarrow \boldsymbol{x}_3$. This helps to perform multiple sequence alignment faster and, more importantly, keeps the information about the states that are used and their ordering, while removing the information about the duration of each state, which is irrelevant.

We then perform multiple sequence alignment on the sequences $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_L$ by maximizing the sum-of-pairs score [29] using the Frank-Wolfe algorithm proposed in [4]. The output is a remapping, $\delta(\boldsymbol{u}_\ell)$, of each sequence $\boldsymbol{u}_\ell$ to a global common template with P slots (in the experiments, we let P be twice the length of the longest sequence). We then generate the final sequence of key-states by voting on each slot in the aligned results. More specifically, for each slot p in aligned sequences, $\delta_p(\boldsymbol{u}_1), \ldots, \delta_p(\boldsymbol{u}_L)$, if the total number of occurrences of the state with the majority vote is bigger than a threshold, we keep the state in the final procedure summary. Otherwise, the result would be empty and will be removed in the final procedure. To generate procedures of different lengths, corresponding to different levels of granularity, we select the voting threshold in order to achieve the desired length.

Once we obtain the final sequence of key-states (each key-state now corresponds to a key-step), we use the assignments found by our optimization to localize each key-step by finding segments assigned to it. The unassigned segments would correspond to background activities.

3.4. Theoretical Guarantees

The greedy algorithm has been shown theoretically to obtain near optimal solutions, when the set function is submodular or approximately submodular [31, 34]. Specifically, $f(\cdot)$ is ϵ-approximately submodular if there exists a submodular function $g(\cdot)$ so that $(1-\epsilon)g(\mathcal{S}) \leq f(\mathcal{S}) \leq (1+\epsilon)g(\mathcal{S})$ for all $\mathcal{S}$. We show that, under certain conditions on transition probabilities, our proposed objective

Dataset	domain	# frames	# videos per task	# tasks	# key-steps per task	foreground ratio	avg video length (sec)	total duration (hr)
[3]	mostly cooking	138,780	5	25	7.7	0.59	221.9	7.7
Breakfast [33]	cooking	1,086,560	50	10	5.1	0.87	137.5	20.0
Inria [4]	various	769,443	30	5	7.1	0.44	178.8	7.4
ProceL	**various**	**4,899,259**	**60**	**12**	**8.3**	**0.63**	**251.5**	**47.3**

Table 1: Comparison of video datasets for procedure learning.

function, which is monotone, is approximately submodular, hence, Algorithm 1 has performance guarantees. The proof of the following is similar to our analysis in [28], with the addition that the sum of approximately submodular functions is approximately submodular.

Theorem 1 *Consider the optimization in (8) with $f(\mathcal{S})$ defined in (9). Assume there exists $\epsilon \in [0, 1)$ such that each log transition probability $q_{i,i'}$ can be written as $q_{i,i'} = \bar{q}_{i'}\psi_{i,i'}$, for some $\bar{q}_{i'}$ and $\psi_{i,i'} \in [1-\epsilon, 1+\epsilon]$. Then, for all values of $\beta \geq 0$, our proposed $f(\mathcal{S})$ is ϵ-approximately submodular. Moreover, the value of f for the solution of the greedy algorithm with budget k is at most $1 - 1/e - O(k\epsilon)$ away from the global optimum of (8).*

4. ProceL Dataset

We present the new multimodal Procedure Learning (ProceL) dataset for research on instructional video understanding. We have collected and annotated 47.3 hours of videos from 720 video clips across 12 diverse tasks, with about 60 videos per task. Out of the twelve tasks, five are the same as the ones presented in the Inria instructional video dataset [4], i.e., *change tire*, *perform CPR*, *make coffee*, *jump car* and *repot plant*. We have expanded these data by including additional 30 videos per task and expanded the set of key-steps to include necessary subactivities to achieve a task. To have a dataset that captures variations of real-world tasks, we have included 7 new tasks: *set up Chromecast*, *assemble clarinet*, *replace iPhone battery*, *tie a tie (Windsor knot)*, *replace toilet*, *make peanut butter jelly sandwich* and *make smoked salmon sandwich*. Our dataset includes detail-oriented tasks, such as 'tie a tie' or 'assemble clarinet', where different key-steps are visually similar, as well as tasks that do not involve interacting with physical objects, such as 'set up Chromecast' in which some steps involve interacting with a virtual environment.

We obtained the videos using YouTube, preferring the ones that clearly show the key-steps required to complete the task and those that include spoken instructions. We trimmed the beginning and end of each video if it included content unrelated to the task, such as product reviews. For each task, we first built a dictionary, which is the set of key-steps necessary to perform the task, e.g., the dictionary of 'install Chromecast' includes {'unpack package', 'download Chromecast app', 'plugin Chromecast power', ..., 'check ready to cast'}. We then annotated each video by localizing all segments during which each key-step in the dictionary has been performed. Figure 1 shows the annotations of six videos from 'perform CPR' and 'replace iPhone battery'. Table 1 shows the comparison[2] of ProceL with other datasets. For the purpose of future research, we also collected spoken instructions generated by Automatic Speech Recognition (ASR) on YouTube. Since the transcripts are noisy, e.g., containing misspelling of words or missing periods, we have corrected the ASR errors.

In addition to various procedure learning tasks, ProceL would be suitable for video-language research as well as weakly-supervised action and object recognition in video.

5. Experiments

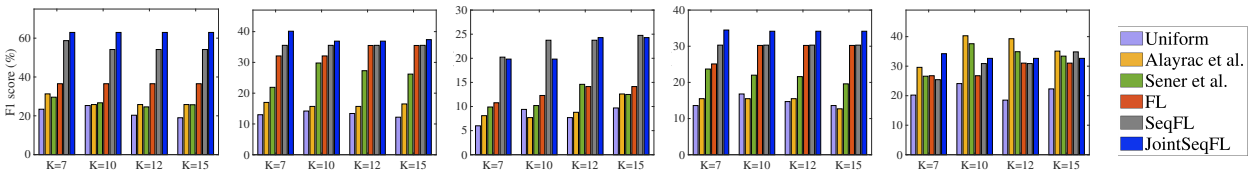
In this section, we evaluate the performance of our procedure learning framework. We perform experiments on the Inria instructional dataset [4] and our new ProceL dataset.

Algorithms and Baselines. We compare our method, Joint Sequential Facility Location (JointSeqFL), with Alayrac et al. [4] and Sener et al. [5] as two state-of-the-art methods for unsupervised procedure learning. To demonstrate the effectiveness of using the dynamic model of data and the joint summarization setting for procedure learning, we compare with the standard Facility Location (FL), which is a single video summarization and does not use dynamic model, and the Sequential Facility Location (SeqFL) [28], which is a single video summarization and uses the dynamic model. We run these two methods on each video individually and align sequences of representatives using multiple sequence alignment [29, 4] Moreover, we use the Uniform baseline, where we distribute key-step assignments uniformly over all segments in each video.

For the experiments, for our method across all tasks, we set $M = 50$, $k = 15$, $\beta = 0.01$ for Inria and $M = 30$, $k = 15$, $\beta = 0.005$ for ProceL. We use pmtk3 toolbox to fit an HMM to videos of each task, where emission probability for each state is a Gaussian. In the supplementary materials, we report the statistics of the ProceL dataset for each task and show qualitative plots of annotations in ProceL.

Feature Extraction. We use [35] to segment each video into superframes, which will be the units of summarization. To have a fair comparison with the state of the art, we extract and use the same features as in Alayrac et. al. [4], which are 3000-dimensional vectors composed of concatenation of 1000-dimensional bag of words appearance features obtained via VGG16 network and 2000-dimensional

[2]We report the statistics of the Breakfast dataset [33] without considering different camera views. We do not compare with YouCookII dataset [16] as it does not have a common dictionary of key-steps for each task.



(a) perform cpr (b) make coffee (c) jump-start car (d) repot plant (e) change tire

Figure 4: F1 scores of different algorithms on the Inria Instructional dataset for different values of procedure length, K.

Procedure length	$K=7$	$K=10$	$K=12$	$K=15$
Inria				
Uniform	15.3	17.9	14.8	15.4
Alayrac et al.	20.3	21.0	21.0	20.5
Sener et al.	22.3	25.2	24.6	23.5
FL	26.3	27.6	29.5	29.5
SeqFL	32.6	34.3	34.0	35.8
JointSeqFL	**38.3**	**37.3**	**38.2**	**38.3**
ProceL				
Uniform	11.0	13.4	13.3	12.8
Alayrac et al.	11.5	12.4	12.8	12.4
JointSeqFL	**27.2**	**28.3**	**28.9**	**29.8**

Table 2: Average F1 scores (%) of different algorithms on the Inria and ProceL datasets for different values of procedure length, K.

bag of words motion features obtained using histogram of optical flows (HOF). For HOF features of each superframe, we use [36], where we set the maximum trajectory length to the length of the superframe. To fit HMM, we reduce the dimension of features via PCA to 300 and 200 on Inria and ProceL, respectively. Supplementary materials contains more details about the feature extraction.

Evaluation Metrics. We use the same evaluation metric as in Alayrac et al. [4] and Sener et al. [5]. More specifically, we first find a one-to-one global matching between the discovered and the ground-truth key-steps across all videos of the same task using the Hungarian algorithm. We then compute the precision, recall and the F1-score, where the precision is the ratio of the total number of correctly localized key-steps to the total number of discovered key-steps across videos. The recall is the ratio of the total number of correctly localized key-steps to the total number of ground truth key-steps across videos. The F1-score is the harmonic mean of the precision and recall and is between 0 and 1. We also compute the Jaccard index, which is intersection over union between discovered and ground-truth key-steps.

Results. Table 2 shows the average F1 scores of different algorithms on the Inria and ProceL datasets as a function of the procedure length $K \in \{7, 10, 12, 15\}$. Notice that on both datasets, our method obtains the best result for all values of K, e.g., achieving the F1 score of 38.3% and 27.2% on Inria and ProceL, respectively, for $K = 7$.

– As the results on Inria show, the three summarization methods perform significantly better than the two state-of-the-art unsupervised procedure learning algorithms, which demonstrates the importance of summarization and subset selection as effective tools for procedure learning.

– Notice that SeqFL and JointSeqFL, which both use the dynamic model, perform better than FL, which treats data points independently, showing the effectiveness of using the transition dynamics in summarization.

– On the other hand, JointSeqFL, which jointly summarizes videos, outperforms SeqFL, which summarizes each video independently, demonstrating the importance of using the common structure of key-steps across instructions of the same task. Another possible limitation of individually summarizing videos and then aligning the results is that since representatives for different videos could be different, alignment leads to putting all representatives across all sequences in the common template with a only one or very few votes, leading to less meaningful aligned result.

– Notice that the performances of all methods on ProceL decrease, since the new 7 tasks such as 'set up Chromecast' or 'replace iPhone battery' are more challenging with more steps than the five common tasks with Inria such as 'perform CPR' or 'change tire'.

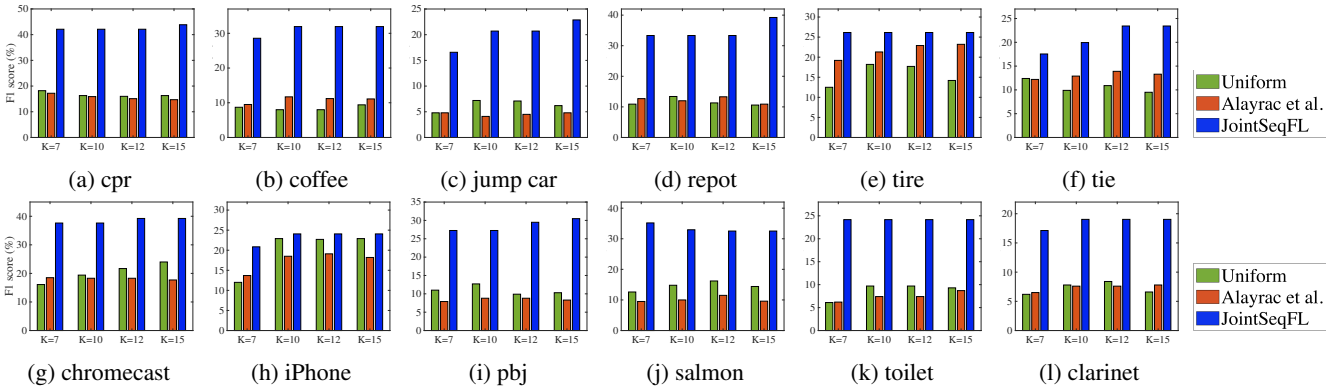
	Inria	ProceL
JointSeqFL ($\beta = 0$)	34.7	27.0
JointSeqFL	**37.3**	**28.3**

Table 3: Effect of using state transitions in summarization on F1 score.

Figures 4 and 5 show the F1 score of different algorithms for each task on Inria and ProceL, respectively.

– On all tasks, except 'change tire' in Inria, summarization-based algorithms perform significantly better than the state of the art, with 'perform CPR' having the most improvement, increasing the F1 score of from 31.3% and 17.2% (via Alayrac et al.) to 62.9% and 42.1% (via JointSeqFL), respectively, on Inria and ProceL. This is due to the fact that 'perform CPR' in both datasets has the largest number of repeated steps, in particular, alternating between 'give compression' and 'give breath' key-steps multiple times. Given the clear dynamic of the task, our joint sequential subset selection method discovers the key-steps more effectively.

– For 'change tire', Alayrac et al. performs better than other algorithms on Inria for $K = 10$ and $K = 12$. In 'change tire', some of the key-steps are more distinguishable by speech than visual data, e.g., 'unscrew lug nuts' and 'screw lug nuts' are visually similar, yet distinct via language. Thus, Alayrac et al. that first takes advantage of speech data to form the sequence of key-steps and then



(a) cpr (b) coffee (c) jump car (d) repot (e) tire (f) tie

(g) chromecast (h) iPhone (i) pbj (j) salmon (k) toilet (l) clarinet

Figure 5: F1 scores of different algorithms on the ProceL dataset for different values of the procedure length, K.

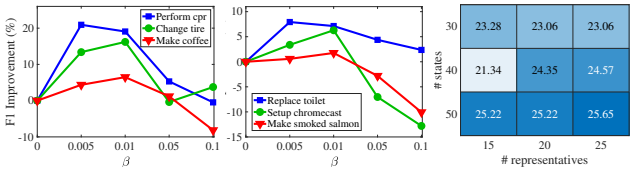


Figure 6: F1 score improvement as a function of β on three tasks from Inria for $K = 7$ (left) and ProceL for $K = 10$ (middle). Effect of the number of states, M, and representatives, k, on the Jaccard index for the task 'perform CPR' in ProceL (right).

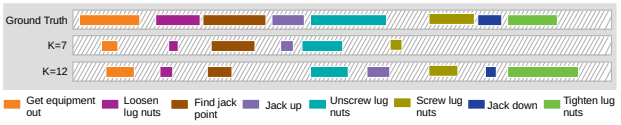


Figure 7: Ground-truth annotation and recovered key-steps by our method for a video of the task 'change tire' with two different procedure lengths, $K = 7$ and $K = 12$. More key-steps are discovered by our method as K increases.

localizes the key-steps in videos, is expected to perform better. However, the performance of this method significantly decreases on ProceL, which contains more videos with many videos having more noisy language descriptions.

Effect of Hyperparameters. Table 3 compares the performance of our method on Inria and ProceL for $K = 10$, with when $\beta = 0$, showing the effectiveness of using the dynamic model. On ProceL the performance gap is smaller, since we have more videos which allows the joint summarization to compensate for the lack of the dynamic model. Figure 6 shows the improvement with respect to $\beta = 0$ on three tasks from Inria and ProceL. Notice that the performance improves as β increases from zero, with 'perform CPR' in Inria and 'replace toilet' in ProceL having the largest F1 score improvement. On the other hand, when β becomes sufficiently large, the performance of 'coffee', 'Chromecast' and 'smoked salmon' decreases with respect to $\beta = 0$, as we overemphasize on the dynamic potential and ignore the encoding. The right plot in Figure 6 shows the stability of the Jaccard index obtained by our algorithm on ProceL as a function of the number of hidden states of HMM, M, and the number of representative states, k.

Qualitative Results. Figure 1 shows the qualitative results of our method for discovery and localization of key-steps from videos of 'perform CPR' (top) and 'replace iPhone battery' (bottom) on ProceL. Notice that for 'CPR', our method discovers and localizes all ground-truth key-steps, except one in the beginning ('call 911' step). For 'replace iPhone battery', while we capture most of the key-steps, we have more miss-localizations, as the task is more challenging than 'CPR', having more key-steps that are also visually similar. Finally, Figure 7 shows the annotations and recovered key-steps by our method for two values of $K = 7$ and $K = 12$ for one video from the task 'change tire'. Notice that increasing K, we effectively recover more ground-truth key-steps (here 'jack down' and 'tighten lug nuts') compared to smaller K. Also, for a given desired procedure length K, we may recover a smaller number of key-steps (as shown in the figure). This is because once we run alignment on the sequences of representatives, to obtain the final procedure, we apply a voting on the aligned results and choose the voting count threshold so that the length of the obtained procedure is at most the desired procedure length, however, no threshold may exactly achieve the length K.

6. Conclusion

We developed a joint dynamic summarization method and a fast greedy algorithm for unsupervised procedure learning. Our method handles repeated key-steps, background and missing or additional key-steps in videos. We presented ProceL, a new multimodal dataset for procedure learning. We showed our method significantly improves the state of the art performance and showed the effectiveness of summarization tools, in general, for procedure learning.

Acknowledgements

This work is partially supported by DARPA Young Faculty Award (D18AP00050), NSF (IIS-1657197), ONR (N000141812132) and ARO (W911NF1810300). E. Elhamifar would like to thank Guandong Liu, Yuan Yu and Maria C. De Paolis Kaluza for their help in collection and annotation of the ProceL dataset.

References

[1] Yezhou Yang, Yi Li, Cornelia Fermüller, and Yiannis Aloimonos, "Robot learning manipulation action plans by" watching unconstrained videos from the world wide web.," *AAAI*, 2015. 1

[2] Nina Mishra, Ryen W. White, Samuel Ieong, and Eric Horvitz, "Time-critical search," *International ACM SIGIR conference on Research & development in information retrieval*, 2014. 1

[3] Ozan Sener, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena, "Unsupervised semantic parsing of video collections," *IEEE International Conference on Computer Vision*, 2015. 1, 2, 6

[4] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant A. Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien, "Unsupervised learning from narrated instruction videos," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2, 3, 5, 6, 7

[5] Fadime Sener and Angela Yao, "Unsupervised learning and segmentation of complex activities from video," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 6, 7

[6] De-An Huang, Shyamal Buch, Lucio Dery, Animesh Garg, Li Fei-Fei, and Juan Carlos Niebles, "Finding it?: Weakly-supervised reference-aware visual grounding in instructional videos," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1

[7] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy, "What's cookin'? interpreting cooking videos using text, speech and vision," *NAACL*, 2015. 1, 2

[8] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, , Tingwu Wang, Sanja Fidler, and Antonio Torralba, "Virtualhome: Simulating household activities via programs," *IEEE Conference on computer Vision and Pattern Recognition*, 2018. 1

[9] Shoou-I. Yu, Lu Jiang, and Alexander Hauptmann, "Instructional videos for unsupervised harvesting and learning of action examples," *ACM International Conference on Multimedia*, 2014. 2

[10] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles, "Connectionist temporal modeling for weakly supervised action labeling," *European Conference on Computer Vision*, 2016. 2

[11] Alexander Richard, Hilde Kuehne, and Juergen Gall, "Weakly supervised action learning with RNN based fine-to-coarse modeling," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[12] Hilde Kuehne, Alexander Richard, and Juergen Gall, "Weakly supervised learning of actions from transcripts," *Computer Vision and Image Understanding Journal*, 2017. 2

[13] Alexander Richard, Hilde Kuehne, and Juergen Gall, "Action sets: Weakly supervised action segmentation without ordering constraints," *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[14] Chloe Kiddon, Ganesa T. Ponnuraj, Luke Zettlemoyer, and Yejin Choi, "Mise en place: Unsupervised interpretation of instructional recipes," *Conference on Empirical Methods in Natural Language Processing*, 2015. 2

[15] De-An Huang, Joseph J. Lim, Li Fei-Fei, and Juan Carlos Niebles, "Unsupervised visual-linguistic reference resolution in instructional videos," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[16] Luowei Zhou, Chenliang Xu, and Jason J. Corso, "Towards automatic learning of procedures from web instructional videos," *AAAI*, 2018. 2, 6

[17] Rameswar Panda and Amit K. Roy-Chowdhury, "Collaborative summarization of topic-related videos," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[18] Wen-Sheng Chu, Yale Song, and Alejandro Jaimes, "Video co-summarization: Video summarization by visual co-occurrence," *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[19] Ehsan Elhamifar, Guillermo Sapiro, and Shankar S. Sastry, "Dissimilarity-based sparse subset selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 2

[20] Ehsan Elhamifar and Maria C. De Paolis Kaluza, "Online summarization via submodular and convex optimization," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[21] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha, "Diverse sequential subset selection for supervised video summarization," *Neural Information Processing Systems*, 2014. 2

[22] Raja H. Affandi, Alex Kulesza, and Emily B. Fox, "Markov determinantal point processes," *Conference on Uncertainty in Artificial Intelligence*, 2012. 2

[23] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman, "Video summarization with long short-term memory," *European Conference on Computer Vision*, 2016. 2

[24] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic, "Unsupervised video summarization with adversarial LSTM networks," *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[25] Ehsan Elhamifar, Guillermo Sapiro, and Rene Vidal, "Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery," *Neural Information Processing Systems*, 2012. 2

[26] Ehsan Elhamifar, Guillermo Sapiro, and Rene Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 2

[27] Ehsan Elhamifar and Maria C. De Paolis Kaluza, "Subset selection and summarization in sequential data," *Neural Information Processing Systems*, 2017. 2, 5

[28] Ehsan Elhamifar, "Sequential facility location: Approximate submodularity and greedy algorithm," *International Conference on Machine Learning*, 2019. 2, 6

[29] Lusheng Wang and Tao Jiang, "On the complexity of multiple sequence alignment," *Journal of Computational Biology*, vol. 1, no. 4, 1994. 3, 5, 6

[30] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, 1978. 4

[31] Andreas Krause and Daniel Golovin, "Submodular function maximization," Cambridge University Press, 2014. 4, 5

[32] Richard Bellman, *Dynamic Programming*. Princeton, NJ: Princeton University Press, 2003. 5

[33] Hilde Kuehne, Ali Arslan, and Thomas Serre, "The language of actions: Recovering the syntax and semantics of goal-directed human," *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 6

[34] Thibaut Horel and Yaron Singer, "Maximizing approximately submodular functions," *Neural Information Processing Systems*, 2016. 5

[35] Michale Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool, "Creating summaries from user videos," *European Conference on Computer Vision*, 2014. 6

[36] Heng Wang and Cordelia Schmid, "Action recognition with improved trajectories," *International Conference on Computer Vision*, 2013. 7