How Local Is the Local Diversity? Reinforcing Sequential Determinantal Point Processes with Dynamic Ground Sets for Supervised Video Summarization

Yandong Li[1,2](✉), Liqiang Wang[1], Tianbao Yang[3], and Boqing Gong[4]

[1] University of Central Florida, Orlando, FL, USA
lyndon.leeseu@outlook.com
[2] Jiulong Lake Campus, Southeast University,
Nanjing 211189, Jiangsu Province, P.R. China
[3] University of Iowa, Iowa City, IA, USA
[4] Tencent AI Lab, Seattle, WA, USA

Abstract. The large volume of video content and high viewing frequency demand automatic video summarization algorithms, of which a key property is the capability of modeling diversity. If videos are lengthy like hours-long egocentric videos, it is necessary to track the temporal structures of the videos and enforce local diversity. The local diversity refers to that the shots selected from a short time duration are diverse but visually similar shots are allowed to co-exist in the summary if they appear far apart in the video. In this paper, we propose a novel probabilistic model, built upon SeqDPP, to dynamically control the time span of a video segment upon which the local diversity is imposed. In particular, we enable SeqDPP to learn to automatically infer *how local the local diversity is supposed to be* from the input video. The resulting model is extremely involved to train by the hallmark maximum likelihood estimation (MLE), which further suffers from the exposure bias and non-differentiable evaluation metrics. To tackle these problems, we instead devise a reinforcement learning algorithm for training the proposed model. Extensive experiments verify the advantages of our model and the new learning algorithm over MLE-based methods.

1 Introduction

The Internet age has come to such a new phase that high-definition videos are both ubiquitous and dominant in the IP traffic featured by the boom of video sharing websites, online movies and television shows, and the emerging live video streaming services. Some statistics indicate that about 300 h of video are uploaded to YouTube per minute and more than 500 million hours of video are watched on YouTube daily. Such a large volume of video content and high viewing frequency demand automatic video summarization algorithms. By distilling important events from the original video and condensing them to a short

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11212, pp. 156–174, 2018.
https://doi.org/10.1007/978-3-030-01237-3_10

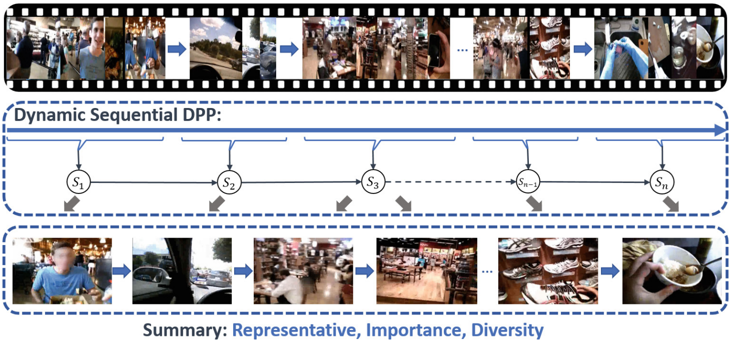

Fig. 1. Dynamic Sequential DPP (DySeqDPP) for video summarization

video clip (or a story board, text description, etc.), video summarization has a great potential in many real-world applications.

Video summarization has been one of the basic research areas in the fields of computer vision and multimedia for decades [2]. A variety of techniques have been proposed for different scenarios of video summarization. In general, a good video summary is supposed to describe main events [3–5] happened in the video and meanwhile remove the video shots that are redundant [6,7] and/or unimportant [8,9].

We consider video summarization as a *diverse* subset selection problem: given a video that can be seen a collection of shots, the goal is to select a subset from the collection to summarize the whole video. This view opens the door for supervised learning approaches to video summarization [1,10–13] that fit subset selection models to the video summaries annotated by users. Unlike the conventional unsupervised video summarization methods [3–5,7–9,14,15], the supervised ones implicitly infer users' intentions and summarization criteria as opposed to domain experts' handcrafting.

In the supervised video summarization models, a key factor they are supposed to encompass is the diversity of the selected subset of video shots. This is often imposed by submodularity [10,16] and determinant [1,11,17]. When a video sequence is short, **global diversity** over the whole sequence seems like a natural choice [10,11].

However, if the videos are lengthy like the egocentric videos that are often hours long, it is necessary to track the temporal structures of the videos and enforce **local diversity** instead [1,18]. The local diversity refers to that the shots selected from a short time duration are diverse but visually similar shots are allowed to co-exist in the summary if they appear far apart in the video. Consider a video sequence that is about "leaving home for shopping in the morning and then coming back home to have lunch". Although the video shots of the "home" scene in the morning may be similar to those at noon, the summary should contain some shots of both in order to make the summary a complete story carried by the video.

In this paper, we are mainly interested in summarizing extremely lengthy (e.g., egocentric) videos and, accordingly, models that are capable of observing the **local diversity**. Among the existing works, sequential determinantal point process (SeqDPP) [1] and dppLSTM [19] both account for the temporal dynamics of the videos. However, neither of them explores *"how local"* the local diversity should be. Take the SeqDPP for instance, it requires users to manually partition the video into disjoint segments of the same length and then impose diversity both within each of them and between adjacent segments, locally. There is no guiding principle about how to best partition a video sequence into such segments. Besides, it could be sub-optimal to make the segments of the same length because different types of events often unroll at distinct frame rates. The same snags exist in dppLSTM.

We propose to improve the SeqDPP model [1] by a latent variable that dynamically controls the time span of a segment upon which the local diversity is then defined in the form of a conditional DPP. In other words, we enable SeqDPP to learn to automatically infer *how local the local diversity is* in the input video. Figure 1 illustrates our main idea. Given an input video shown on the top panel, our dynamic SeqDPP seeks the appropriate and possibly different lengths of the segments (cf. the middle panel) from which it selects video shots (the bottom panel) and places them on a story board or links them into a short video clip as the summary of the video.

Another contribution of this paper is a novel reinforcement learning algorithm for the proposed dynamic SeqDPP (DySeqDPP). While DySeqDPP seems like a straightforward extension to the vanilla SeqDPP, it is less obvious how to efficiently train the model. The DPPs [20] and its variants (e.g., SeqDPP [1], dppLSTM [19], and SH-DPP [17]) are almost all trained by the hallmark maximum likelihood estimation (MLE) except for the large-margin DPP [21] and Bayesian DPP [22]. However, it is often difficult to maximize the likelihood of a sequential model with latent variables; gradient ascent fails to track the statistical structure, and the EM algorithm [23] becomes involved and inefficient unless one assumes special compositions of a sequential model [24].

In light of these challenges, we instead provide a reinforcement learning perspective for understanding SeqDPPs. The proposed DySeqDPP is used as a policy by an agent to interact with the environment—the input video. Accordingly, we train this DySeqDPP model by policy gradient descent [25]. Not only we do not have to explicitly deal with the latent variables, but also we benefit from the flexible reward functions in policy gradient descent—we can bridge the training and validation phases of the summarizer by defining the reward function as some evaluation metric(s).

We evaluate this dynamic SeqDPP model on standard video summarization datasets. Extensive results show that it significantly outperforms competing baselines especially the vanilla SeqDPP, verifying the necessity of dynamically determining how local the local diversity is. The rest of the paper is organized as follows. Section 2 discusses some related existing video summarization works. After that, we describe our dynamic SeqDPP and the reinforcement learning algorithm in Sect. 4. We report empirical results in Sect. 5 and then conclude the paper by Sect. 6.

2 Related Work

2.1 Video Summarization

Different algorithms for automatic video summarization are generally designed by the same principles. Those informative guidelines contain three main factors: (1) individual interestingness or relevance [8,9], which means selecting frames/shots that are important in the video; (2) representativeness [3–5], which means the summary should contain the main event of the videos; (3) collective diversity or coverage [6,7], which is to reduce redundant frames/shots without losing much information. These factors are used in most of the existing works. Next, we review the representative approaches in two common classes, unsupervised and supervised video summarization.

Unsupervised Video Summarization: A variety of prior works is designed based on basic visual quality like low-level appearance and motion cues [3–9,14,15,26]. Graph models are utilized for event detection in some approaches [5,26]. In general, the criteria applied in those methods for making decisions about including or excluding shots are devised by the system developers empirically. Besides, some approaches leverage Web images for video summarization based on the assumption that the static Web pictures tend to contain information of interest to people, so the Web images reveal user-oriented importance selecting video shots/frames [4,27–29].

Supervised Video Summarization: Recently, several explorations on supervised video summarization have been exerted for various goals [1,8–13,17–19,30]. They achieve superior performance over the traditional unsupervised clustering algorithms. Among them, Gygli *et al.* try to add some supervised flavor to optimize mixture objectives with learning each criterion's weight [10,12]. A hierarchical model has been proposed to learn with few labels, and it is optimized to generate video summary containing interesting objects [30]. Egocentric videos [31] can be compacted with importance of people and objects [8]; on the other hand, Zheng *et al.* explicitly consider how one sub-event leads to another in order to provide a better sense of story for those kinds of videos [9]. Meanwhile, Yao *et al.* propose a pairwise deep ranking model to highlight video segments of first-person videos [32]. In conclusion, supervised methods are capable of utilizing the intentions of users about what a qualified video summary is rather than designing the systems only relying on the experts' own perspective.

Besides, as a powerful diverse subset selection model, the determinantal point process (DPP) has been widely used for video summarization. For instance, Gong *et al.* propose the first supervised video summarization method [1] (SeqDPP) as far as we know, it models local diversity to capture the temporal information of videos rather than modeling global diversity. Combining long short-term memory (LSTM) with DPPs has been studied in [19] to model the variable-range temporal dependency and diversity among video frames at the same time. Effort has been spent to study transferring summary structures from annotated videos to unseen test videos in [11]. Sharghi *et al.* explore the query-focused video summarization in [17,18]. Large margin separation principle has been leveraged for DPPs to estimate parameters in [13].

We will provide more details of DPPs and SeqDPP in Sects. 3.1 and 3.2.

Reinforcement learning (RL) provides a unified solution to both problems above. The REINFORCE algorithm [38] is utilized to train recurrent neural network [33]. Rennie *et al.* borrow ideas from [33] in the image captioning task and obtain very promising results [39]. We note that the use of RL in those contexts is icing on the case in the sense that, while RL boosts the results to some degree, the MLE is still applicable. For our DySeqDPP model, however, RL becomes a necessary choice because it is highly involved to handle the latent variables in DySeqDPP by MLE.

3 Background: DPP and SeqDPP

We briefly review the determinantal point process (DPP) and the sequential DPP (SeqDPP) in this section. It will become clear soon how the former promotes diversity in the selected subsets and the latter enables local diversity.

3.1 DPPs

A discrete DPP defines a distribution over the subsets of a ground set and assigns high probability to a subset if its items are diverse from each other. The notion of diversity is induced by a kernel matrix whose entries can be understood as pairwise similarities between the items. The more similar two items are, the less likely they co-occur in a subset sampled from the DPP.

More concretely, given a ground set $\mathcal{Y} = \{1, 2, \ldots, \mathsf{N}\}$ of N items, let $\boldsymbol{K} \in \mathbb{R}^{\mathsf{N}\times\mathsf{N}}$ be a symmetric positive semidefinite matrix, called the kernel of DPP. It measures pairwise similarities between the N items. A distribution over a random subset $Y \subseteq \mathcal{Y}$ is a DPP, if for every $\boldsymbol{y} \subseteq \mathcal{Y}$ we have

$$P_{dpp}(\boldsymbol{y} \subseteq Y; \boldsymbol{K}) = \det(\boldsymbol{K}_{\boldsymbol{y}}), \tag{1}$$

where $P_{dpp}(\cdot)$ is the probability of an event, $\boldsymbol{K}_{\boldsymbol{y}}$ denotes a squared submatrix of $\boldsymbol{K}$ with rows and columns indexed by $\boldsymbol{y}$, and $\det(\cdot)$ is the determinant of a matrix. All the eigenvalues of the kernel matrix $\boldsymbol{K}$ are between 0 and 1. Since $P(i, j \in Y; \boldsymbol{K}) = K_{ii}K_{jj} - K_{ij}^2$, *i.e.*, the probability of any two items i, j co-existing in the random subset Y is discounted by their similarity K_{ij}. In other words, the subsets whose items are less similar to each other are assigned higher probabilities than the other subsets.

L-Ensemble. In practice, it is often more convenient to use the so-called L-ensemble DPP that directly assigns atomic probabilities to all the possible subsets of the ground set. Let $\boldsymbol{L}$ denote a symmetric positive semidefinite matrix in $\mathbb{R}^{\mathsf{N}\times\mathsf{N}}$. The L-ensemble DPP draws a subset $\boldsymbol{y} \subseteq \mathcal{Y}$ with probability

$$P_L(Y = \boldsymbol{y}; \boldsymbol{L}) = \det(\boldsymbol{L}_{\boldsymbol{y}})/\det(\boldsymbol{L} + \boldsymbol{I}), \tag{2}$$

where $\boldsymbol{I}$ is an identity matrix. The corresponding marginal kernel that defines the marginal probability in (1) is given by $\boldsymbol{K} = \boldsymbol{L}(\boldsymbol{L} + \boldsymbol{I})^{-1}$.

Conditional DPP. One of the appealing properties of DPP is that there exists an analytic form of its conditional distribution. For any $\boldsymbol{y}_1 \subseteq \mathcal{Y}$ and $\boldsymbol{y}_0 \subseteq \mathcal{Y}$, $\boldsymbol{y}_1 \cap \boldsymbol{y}_0 = \emptyset$,

$$P_L(Y = \boldsymbol{y}_1 \cup \boldsymbol{y}_0 | \boldsymbol{y}_0 \subseteq Y; \boldsymbol{L}) = \det(\boldsymbol{L}_{\boldsymbol{y}_1 \cup \boldsymbol{y}_0}) / \det(\boldsymbol{L} + \boldsymbol{I}_{\mathcal{Y} \setminus \boldsymbol{y}_0}), \tag{3}$$

where $\boldsymbol{I}_{\mathcal{Y} \setminus \boldsymbol{y}_0}$ is a matrix with ones in the diagonal entries indexed by $\mathcal{Y} \setminus \boldsymbol{y}_0$ and zeros everywhere else. Kulesza and Taskar have written an excellent tutorial about DPPs [40].

3.2 Sequential DPPs

A sequential DPP (SeqDPP) [1] was proposed for supervised video summarization. It adheres to the inherent temporal structure in video sequences, thus overcoming the deficiency of DPPs which treat video frames/shots as randomly permutable items. The main technique is to use the conditional DPPs to construct a Markov chain.

Given a long video sequence $\mathcal{V}$, we partition it into T disjoint yet consecutive short segments $\bigcup_{t=1}^{\mathsf{T}} \mathcal{V}_t = \mathcal{V}$. At the t-th time step, SeqDPP selects a diverse subset of items (e.g., frames or shots), by a variable $X_t \subseteq \mathcal{V}_t$, from the corresponding segment conditioning on the items $\boldsymbol{x}_{t-1} \subseteq \mathcal{V}_{t-1}$ selected from the immediate past segment. This subset selection variable X_t follows a distribution given by the conditional DPP,

$$P_{seq}(X_t = \boldsymbol{x}_t | X_{t-1} = \boldsymbol{x}_{t-1}, \mathcal{V}_t) := P_L(Y_t = \boldsymbol{x}_t \cup \boldsymbol{x}_{t-1} | \boldsymbol{x}_{t-1} \subseteq Y_t; \boldsymbol{L}^t) \tag{4}$$

$$= \det(\boldsymbol{L}^t_{\boldsymbol{x}_t \cup \boldsymbol{x}_{t-1}}) / \det(\boldsymbol{L}^t + \boldsymbol{I}^t_{\mathcal{V}_t}), \tag{5}$$

where $P_L(Y_t; \boldsymbol{L}^t)$ is an L-ensemble with the ground set $\boldsymbol{x}_{t-1} \cup \mathcal{V}_t$. Denote by $\boldsymbol{x}_0 = \emptyset$. The SeqDPP over all the subset selection variables is factorized as

$$P_{seq}(\{X_t = \boldsymbol{x}_t\}_{t=1}^{\mathsf{T}}, \mathcal{V}) = \prod_{t=1}^{\mathsf{T}} P_{seq}(X_t = \boldsymbol{x}_t | X_{t-1} = \boldsymbol{x}_{t-1}, \mathcal{V}_t). \tag{6}$$

Figure 2 illustrates SeqDPP and compares it to the vanilla DPP and Markov DPP [41]. Unlike the vanilla or Markov DPPs which considers the video frames/shots as orderless items, SeqDPP maintains the temporal order among the segments and yet ignores it among the frames/shots within an individual segment, locally. Furthermore, it retains the diversity property for adjacent video segments but not for those that are far apart. Indeed, users may want to keep visually similar video clips in the summary if they are far apart in a lengthy video in order to tell a complete story of the video.

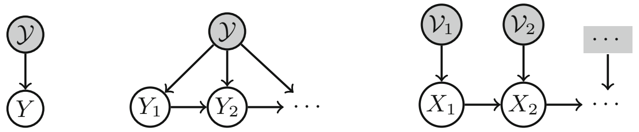

Fig. 2. From left to right: Determinantal point process (DPP) [40], Markov DPP [41], and sequential DPP (SeqDPP) [1]. The ground sets are denoted by the shaded nodes.

3.3 Reinforcement Learning

Consider an agent that takes actions according to some policy to interact with the environment. Following the popular Markov decision process (MDP) formalism, we describe the problem by $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state (s) space and action (a) space, respectively, $P(s_{t+1}|s_t, a_t)$ is a state transition distribution, $R(s_{t+1}; s_t, a_t)$ is a reward the agent receives if it takes action a_t at state s_t and results in state s_{t+1}, and $\gamma \in (0, 1)$ is a discount factor. A policy is denoted by $\pi : \mathcal{S} \mapsto \mathcal{A}$, which is essentially a conditional distribution $\pi(a_t|s_t)$ over the actions given any state. Reinforcement learning aims to find the agent a policy that maximizes the expected total discounted reward $\mathbb{E}_\pi \sum_{i=0}^{\infty} \gamma^i R_{t+i}$ starting from time step t.

4 Reinforcing Dynamic SeqDPPs

We are now ready to present our dynamic SeqDPP (DySeqDPP) along with a reinforcement learning algorithm for estimating the model parameters.

4.1 DySeqDPP

We describe the DySeqDPP model using the MDP formalism $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ so that the corresponding learning algorithm follows naturally. We note that, in addition to the new DySeqDPP, another contribution of this section is the reinforcement learning perspective for understanding SeqDPPs. Under this framework, SeqDPP and DySeqDPP can be seen as two types of stochastic policies.

State s_t at time step t: An information state is about the history of an agent's observations (and rewards) about the environment. It is used to determine what happens next upon an action taken by the agent. In our context, the state $s_t = \{\bigcup_{t'=1}^{t-1} \boldsymbol{x}_{t'}, \mathcal{V}_t\}$ comprises the dynamic partition of the video $\mathcal{V}_t$ at time step t and the generated video summary $\bigcup_{t'=1}^{t-1} \boldsymbol{x}_{t'}$ right before the current step t. One may wonder to alternatively treat all the video segments $\mathcal{V}_1, \cdots, \mathcal{V}_t$ until step t as the state. We contend that it is oppressive and unnecessary to carry them along over time. Instead, the summary of the past conveys similar amount of information by design.

Action a_t at time step t: In DySeqDPP, the agent takes actions (1) to select a subset X_t from the video segment $\mathcal{V}_t$ and (2) to propose the length L_t of the next segment $\mathcal{V}_{t+1}$. The subset selection variable $X_t \subseteq \mathcal{V}_t$ and the partition proposal variable $L_t \in \mathcal{L}$ jointly define the action space. In other words, an action takes the form of $A_t = (X_t, L_t)$ whose realization is denote by $a_t = (\boldsymbol{x}_t, l_t)$. We limit the search of the segment's length to the range of $\mathcal{L} = \{5, 6, \cdots, 15\}$ shots.

Policy π: We let the agent take a stochastic policy in the following manner,

$$\pi(a_t|s_t) = P(\boldsymbol{x}_t, l_t|s_t) = P(\boldsymbol{x}_t|s_t)P(l_t|\boldsymbol{x}_t, s_t), \quad (7)$$

where $P(\boldsymbol{x}_t|s_t)$ is a conditional DPP used to build SeqDPP [1], *i.e.*,

$$P(\boldsymbol{x}_t|s_t) = P(\boldsymbol{x}_t| \cup_{t'=1}^{t-1} \boldsymbol{x}_{t'}, \mathcal{V}_t) := P_L(Y_t = \boldsymbol{x}_t \cup \boldsymbol{x}_{t-1}|\boldsymbol{x}_{t-1} \subseteq Y_t; \boldsymbol{L}^t) \quad (8)$$

and $P(l_t|\boldsymbol{x}_t, s_t)$ is defined as a softmax function,

$$P(l_t|\boldsymbol{x}_t, s_t) = P(l_t| \cup_{t'=1}^{t} \boldsymbol{x}_{t'}, \mathcal{V}_t) := \texttt{softmax}(\boldsymbol{w}_{l_t}^T \phi(\cup_{t'=1}^{t} \boldsymbol{x}_{t'}, \mathcal{V}_t)). \quad (9)$$

There are several points in the above worth clarifying and discussing. First of all, Eq. (7–9) describe the main body of our DySeqDPP model. It improves SeqDPP by the partition proposal variable L_t. It is a latent variable because users annotate summaries of videos without explicitly knowing the boundaries of the local diversities they have in their minds. Secondly, we condition the DPP in Eq. (8) on its immediate past time step ($\boldsymbol{x}_{t-1}$) only instead of the whole history of summaries included in the state s_t. This is due to the same modeling intuition as SeqDPP, *i.e.*, in order to maintain local diversity in the summaries. Thirdly, $\phi(\cdot)$ in Eq. (9) extracts features by max-pooling the representations of all the video shots in the current state s_t as well as the new summary $\boldsymbol{x}_t$ selected according to Eq. (8). This ensures that sufficient information about both the whole past history and the current of the video is supplied to the `softmax` for the agent to predict the appropriate length of the next segment. Last but not the least, $\{\boldsymbol{w}_l, l \in \mathcal{L}\}$ are the model parameters to be learned from the user-annotated summaries. It is important to note that the parameters are not bound to any particular environments/videos at all, so the policy can be generalized to unseen videos, too. We postpone the parameterization of the L-ensemble DPP's kernel L to Sect. 4.2.

State-action value function: Our goal is to learn a policy to maximize the expected total discounted reward the agent receives, called the state-action value function,

$$Q^{\pi}(s_0, a_0) := \mathbb{E}_{\pi}\Big[\sum_{t=0}^{T} g(\gamma, t)R_t|S_0 = s_0, A_0 = a_0\Big], \quad (10)$$

where $g(\gamma, t) \in [0, 1]$ is a discount function and the reward $R_t = R(s_{t+1}; s_t, a_t)$ is a function of the state and action. For video summarization, the reward can be evaluation metrics like precision, recall, or F-score computed between the video shots $\cup_{t'=1}^{t} \boldsymbol{x}_{t'}$ selected by the agent and the user summaries of the video (until the current segment $\mathcal{V}_t$). The total number of time steps the agent can take is T, which satisfies $\sum_{t=0}^{T-1} l_t < |\mathcal{V}|$ and $\sum_{t=0}^{T} l_t \geq |\mathcal{V}|$.

It is import to note that our goal is to maximize the state-action value function at the initial state and action (s_0, a_0) which are fixed to $s_0 = \emptyset$ and $a_0 = (\boldsymbol{x}_0 = \emptyset, l_0 = 10)$ in our experiments. In contrast to conventional setups in reinforcement learning, we do not care about the state-action values at other states because only the initial state gives rise to a whole summary of the video, which is our interest. This insight also suggests a special design of the discount function $g(\gamma, t)$. Instead of using the common practice γ^t, we let it be $g(\gamma, t) = \gamma^{|\mathcal{V}-t|}, \gamma \in (0, 1)$, monotonically increasing with respect to t in order to weigh the reward of the whole summary more than the incomplete summaries at any other time steps.

Those differences highlight the fact that video summarization actually lacks some characteristics of reinforcement learning (e.g., delayed feedback). Hence, we have to customize the MDP formalism in order to match it with the goal of interest. Nonetheless, by casting DySeqDPP as a policy, we can conveniently learn its model parameters by algorithms in reinforcement learning—we employ gradient descent in this paper.

4.2 Policy Gradient Descent for Learning DySeqDPP

We review the model parameters in DySeqDPP before deriving the learning algorithm. We parameterize two conditional distributions in DySeqDPP for the purpose of out-of-sample extension, so that one can readily apply the learned model to unseen test videos. The first is the partition proposal distribution (Eq. (9)) and the second is the conditional DPP (Eq. (8)) at each time step t, whose L-ensemble kernel is constructed as follows,

$$[\boldsymbol{L}^t]_{ij} = \boldsymbol{z}_i^T \boldsymbol{W}^T \boldsymbol{W} \boldsymbol{z}_j, \qquad \boldsymbol{z}_i = \texttt{ReLU}(\boldsymbol{U}\,\texttt{ReLU}(\boldsymbol{V}\boldsymbol{f}_i)) \tag{11}$$

where $\boldsymbol{f}_i$ is the feature representation of video shot i in the ground set $\boldsymbol{x}_{t-1} \cup \mathcal{V}_t$ of the time step t. This feature vector goes through a feedforward network with ReLU activations. Denote by θ the union of the weights of the network $(\boldsymbol{W}, \boldsymbol{U}, \boldsymbol{V})$ and the unknowns $\{\boldsymbol{w}_l, l \in \mathcal{L}\}$ in Eq. (8). We next derive a learning algorithm using the policy gradient descent [42] to estimate the model parameters θ.

Recall that our goal is to maximize the state-action value function at the initial state and action. Denoting by $J \triangleq -Q^{\pi}(s_0, a_0)$, we can minimize it by gradient descent,

$$\nabla_\theta J|_{\theta=\theta_{\text{old}}} = -\mathbb{E}_{\boldsymbol{\tau}\sim\pi(\theta_{\text{old}})}\Big[\sum_{t=1}^{T} g(\gamma, t) R_t \nabla_\theta \log p(\boldsymbol{\tau}; \theta)|_{\theta=\theta_{\text{old}}}\Big] \tag{12}$$

$$\approx -\frac{1}{K}\sum_{k=1}^{K}\Big[\sum_{t=1}^{T_k} g(\gamma, t)\, r_{tk}\, \nabla_\theta \log p(\boldsymbol{\tau}_k; \theta)|_{\theta=\theta_{\text{old}}}\Big] \tag{13}$$

where the last equation is obtained by sampling K trajectories $\{\boldsymbol{\tau}_k\}$ from the policy instantiated by the old parameter θ_{old}, r_{tk} is the reward that the agent receives at time step t of the k-th trajectory, and the first equation is due to the following fact,

$$\nabla_\theta \mathbb{E}_{x\sim\theta}[f(x)]|_{\theta=\theta_{\text{old}}} = \mathbb{E}_{x\sim\theta_{\text{old}}}\Big[\nabla_\theta \log p(x;\theta)|_{\theta=\theta_{\text{old}}} f(x)\Big]. \tag{14}$$

We still need to work out $\nabla_\theta \log p(\boldsymbol{\tau};\theta)$ in Eq. (13). The key is that the state-transition distribution $p(s_{t+1}|s_t,a_t)$ is actually deterministic under our context laid out in Sect. 4 (because the action a_t fully determines the summary $\boldsymbol{x}_t$ and the next segment $\mathcal{V}_{t+1}$, and hence the next state). Therefore, for a trajectory $s_0, a_0, s_1, a_1, \cdots$, we have

$$\nabla_\theta \log p(\boldsymbol{\tau};\theta) = \nabla_\theta \log \Big[p(s_0,a_0)\prod_{t=1}^{T} p(s_t|s_{t-1},a_{t-1})\pi(a_t|s_t;\theta)\Big] \tag{15}$$

$$= \nabla_\theta \sum_{t=1}^{T} \log \pi(a_t|s_t;\theta) = \sum_{t=1}^{T}\Big[\nabla_\theta \log P(\boldsymbol{x}_t|s_t) + \nabla_\theta \log P(l_t|\boldsymbol{x}_t,s_t)\Big] \tag{16}$$

where the first summand of the last equation is the gradient with respect to the parameters of conditional DPP and the second is of the `softmax` (Eq. (9)).

Implementation: Instead of computing the gradients explicitly, one may use the "autodiff" feature of many existing deep learning tools to obtain the gradients. Take PYTORCH (http://pytorch.org) for instance. We may program the following for a trajectory,

$$J(\boldsymbol{\tau};\theta) = -\sum_{t=1}^{T} g(\gamma,t)\, r_t \Big[\log P(\boldsymbol{x}_t|s_t;\theta) + \log P(l_t|\boldsymbol{x}_t,s_t;\theta)\Big],$$

and then use the `backward()` function to automatically compute the gradients followed by calling the `step()` function to do a one-step gradient descent. After that, we sample another trajectory and repeat the procedure until the termination condition.

5 Experiments

We run experiments on three datasets, UTE [8], SumMe [12], and TVSum [43], and compare our approach to several competing baselines.

5.1 The UT Egocentric (UTE) Dataset

Data and Features. UTE [8] contains four egocentric videos, each of which lasts between three and five hours long. It captures daily activities such as shopping in a grocery store, having lunch, working, chatting with friends, meeting with colleagues, etc. In addition to the big variety of content, the videos are also quite challenging due to ego motions—as a result, the views change frequently. The motion blur is more frequent and severe than "third-person" videos. In general, the video shots of an activity are placed in between of blurred frames and nuisance views. Following the experiment protocol of [18], we run four rounds of experiments. In each round, we use two videos for training, one for validation, and the last for testing. We uniformly divide the videos to 5-second shots. From each video frame, we extract 4,096D deep CNN features as the activation of the last fully connected layer of the VGG19 network [44] pretrained on ImageNet [45]. After that, we use PCA to reduce the feature dimension from 4,096D to 512D, followed by max-pooling within each shot in order to have a shot-level feature representation (*i.e.*, $\boldsymbol{f}_i$ in Eq. (11)).

Competing Methods. We mainly compare our approach (DySeqDPP) to the following methods and their variations which, like ours, locally promote diversity in video summaries: SeqDPP [1,9], dppLSTM [19], and uniform sampling (Uniform). We let the methods automatically work out the lengths of the summaries except for the uniform sampling, to which we supply the lengths of the oracles. For SeqDPP, however, the length of each segment has to be manually set. In addition to the 10-shot segments suggested in the original work [1], we also include the results of segments of 5 shots and 12 shots. Finally, we include another comparison by improving the original SeqDPP with our reinforcement learning algorithm. This is implemented by fixing the partition proposal distribution $P(L_t|\boldsymbol{x}_t, s_t)$ as a Dirac delta function $\delta(L_t = l)$, where $l = 10$ is independent of the time steps. Besides, we learn using the reward of the whole summary by setting $g(\gamma, t) = 0$ for $t < T$ and $g(\gamma, T) = 1$, unless specified otherwise.

Evaluation. In the literature, system-generated summaries have been evaluated in a variety of manners including but not limited to user studies [46], percentage of frames overlapped with user summaries [19], bipartite matching based on distances of low-level visual features [18], etc. Arguably, user study is the "gold" standard, but it is extremely time-consuming. In this paper, we instead use the bipartite matching based on a "semantic distance"—pairwise Hamming distance between video shots computed upon the concepts annotated for each shot. This imitates user studies in the sense that the "semantic distance" is strongly correlated with users' perceptions about the difference between a system-generated summary and an actual user's summary. The concepts per video shot are borrowed from an earlier work by Sharghi et al. [18], in which the authors asked users to choose from 54 concepts the ones relevant to a given video shot.

Given two summaries (*i.e.*, a system-generated one and a user summary), we construct a bipartite graph between them with the shots as nodes. A node in one part is connected to all the nodes in the other part with edge weights as the (negative) Hamming distance computed from the per-shot concepts [18]. After that, we find the size of the maximum bipartite matching and divide it by the length of the user (system) summary to obtain the recall (precision).

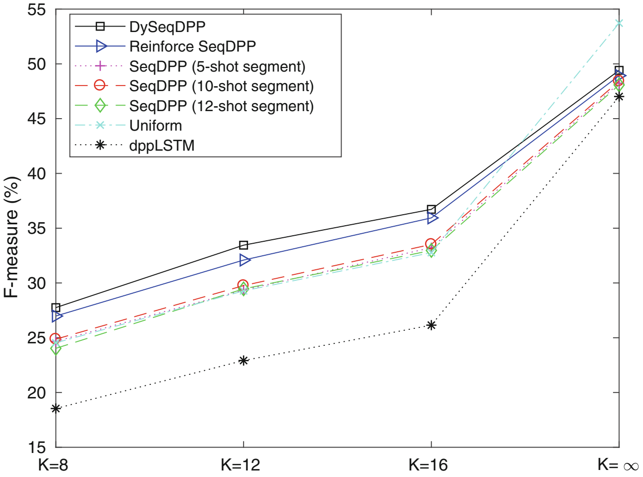

Fig. 3. Comparison results in terms of average F1-score (the higher, the better) for 4 videos in UTE dataset where horizon axis means different K used in local bipartite matching.

Additionally, we improve this metric by removing the edges between the video shots that are more than K time steps away from each other. In other words, if two shots are far away from each other for more than $5K$ seconds, there is no edge between them in the improved evaluation metric.

Comparison Results. Figure 3 reports the results using the above evaluation scheme at $K = 8, 12, 16, \infty$. Each system-generated summary is compared against three user summaries and the corresponding precision, recall, and F-measure scores are averaged to reduce user bias. We can see that the proposed DySeqDPP outperforms the competing methods by a large margin. The SeqDPP trained by our reinforcement learning algorithm ranks the second. These results verify the benefit of understanding the SeqDPPs from the novel reinforcement learning perspective. Moreover, the latent variable for dynamically partitioning the videos into segments also helps. It not only removes the need of handcrafting the segments but also gives rise to superior performance over the equally paced segments.

Another intriguing observation is that there is no significant difference among the results of SeqDPP when we change the sizes of the segments (*i.e.*, 5, 10, and 12 shots). It indicates that one can hardly find an “optimal” length for the equally placing segments of SeqDPP, signifying the need of dynamically partitioning the videos to segments of variable lengths as our DySeqDPP does.

It is a little surprising to see that dppLSTM underperforms uniform sampling. Upon examining the existing works [18, 47] carefully, we find that uniform sampling is actually a very competitive baseline partially because it receives unfair information at inference—length of the oracle summary. Another possible reason is that we did not pre-train the dppLSTM using any additional datasets as done in [19].

Table 1. Comparison results on UTE evaluated by the bipartite matching F1-score ($K = 12$)

Method	$\gamma = 1e^{-20}$	Full $\gamma = 0.2$	Full $\gamma = 0.5$	Full $\gamma = 0.9$	Partial $\gamma = 0.2$	Partial $\gamma = 0.5$	Partial $\gamma = 0.9$	Greedy sample	Pool seg	Pool video
Video 1	29.53	28.96	28.03	29.27	28.83	28.33	28.23	27.76	29.19	30.33
Video 2	31.17	30.67	31.61	30.80	32.53	32.07	30.91	29.24	31.20	31.90
Video 3	46.38	45.79	45.88	42.04	45.20	45.23	44.42	43.56	40.40	43.68
Video 4	26.72	26.93	26.35	26.51	26.07	26.41	27.51	23.81	24.56	24.91
Avg.	33.45	33.08	32.96	32.16	33.15	33.01	32.77	31.09	31.33	32.71

Ablation Study. Besides, we run some ablation studies to test several variations to our approach and illustrate the quantitative results in Table 1. First, instead of sampling K trajectories $\{\boldsymbol{\tau}_k\}$ based on the old policy, we sample the trajectory $\boldsymbol{\tau}$ in a greedy manner, which chooses the subsets with the maximum probability at each step during training. The "Greedy Sample" column in Table 1 indicates that greedy sampling produces worse video summarization results. The reason is that the system can not explore the real environment (video) thoroughly under the greedy sampling strategy.

We also study how the hyper-parameter γ ($\gamma = 1e^{-20}, 0.2, 0.5, 0.9$) influences the model. Specifically, larger γ means we give higher weight to the incomplete summaries at early time steps. Meanwhile $\gamma = 1e^{-20}$ means we just consider the whole video summary at the final time step. The experimental results in Table 1 verify our intuitive assumption that weighing more on the reward of the whole summary is better than on the incomplete summaries at other time steps. In addition, we notice a problem that it is unreasonable to calculate the reward of each time step by comparing the incomplete summary up to the current step with the full user summary (shown in the columns titled "Full $\gamma = 0.2/0.5/0.9$"). To address this problem, we compute the reward by comparing the current system summary with the user summary until this time step, as shown in the column titled "Partial $\gamma = 0.2/0.5/0.9$". The experimental results verify that the latter kind of reward calculation is more reasonable.

Finally, we also study what features work better for predicting l_t. Recall that, for $\phi(\cup_{t'=1}^{t} \boldsymbol{x}_{t'}, \mathcal{V}_t)$, we concatenate the features of the generated video summary until the current time step and the features of the current segment. We test two alternatives. One is pooling the features of this segment only (PoolSeg) and the other is pooling the features of the whole video sequence up to the current segment (PoolVideo). PoolSeg gives rise to worse results than PoolVideo since it lacks the larger context than the current segment only. PoolVideo is a little worse than and certainly incurs more computation cost than $\phi(\cup_{t'=1}^{t} \boldsymbol{x}_{t'}, \mathcal{V}_t)$ because pooling over the video encounters redundant information.

5.2 The SumMe and TVSum Datasets

Experiment Setup. In addition to the egocentric videos, we also test our approach on two other popular datasets for video summarization: **SumMe** [12]

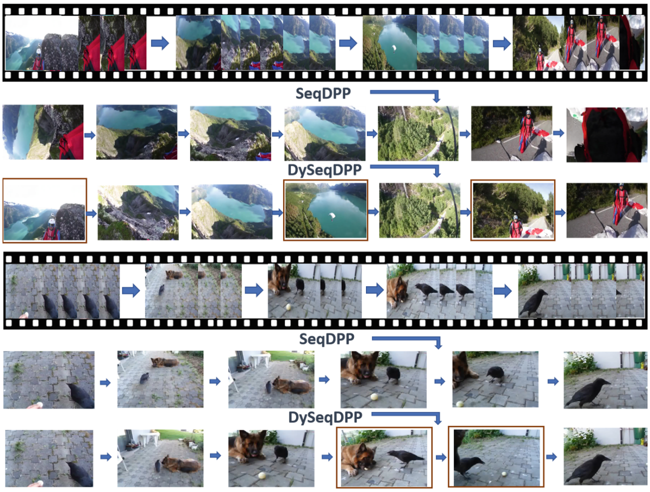

Fig. 4. Generated video summary examples with SeqDPP and DySeqDPP

and **TVSum** [43]. They are both "third-person" video datasets. SumMe consists of 25 consumer videos covering holidays, events, and sports. The lengths of the videos range from about one to six minutes. TVSum contains 50 videos of 10 categories downloaded from YouTube. The videos are one to five minutes in length.

We follow the same experimental setup as dppLSTM [19] in this work. We extract the output (1,024D) of the penultimate layer (pool 5) of GoogLeNet [48] for each video frame. Followed by max-pooling within each shot (15 frames), we get the shot-level feature representation. In our experiments, we train the model with 60% videos of SumMe (TVSum), validate on 20% of the dataset, and test on the remaining 20% videos. We run 10 rounds of experiments with different random splits of the dataset and report both the mean F1-scores and standard errors.

Evaluation. We evaluate the results again by F1-score. However, the precisions and recalls for computing the F1-score are calculated in a different way from the bipartite graph matching earlier. Following by the practice in dppLSTM [19], we first split a video into a set of disjoint temporal scenes (which are usually longer and contain more visual information than the segments and shots used in the UTE dataset) using the KTS approach [49]. We train the model with shot-level feature representations and then use it to obtain shot-level importance scores. Specifically, the importance score of each frame is equal to the score of shots they belong to. We compute the scene-level scores by averaging the scores of frames within each scene and then rank the scenes in the descending order by their scores. In order to generate a video summary, we select the scenes with a duration below a certain threshold (*e.g.*, using the knapsack algorithm as in [43]).

Table 2. Comparison results on video summarization on SumMe and TVsum dataset. The results are evaluated by F1-score, the higher the better.

Dataset	Method	Unsupervised	Canonical
SumMe	Video-MMR [50]	26.6	
	Gygli *et al.* [12]		39.4
	Gygli *et al.* [10]		39.7
	Zhang *et al.* [11]		40.9
	vsLSTM [19]		37.6 ± 0.8
	dppLSTM [19]		38.6 ± 0.8
	SeqDPP [1]		40.8 ± 4.8
	DySeqDPP		$\mathbf{44.3 \pm 2.8}$
TVSum	LiveLight [51]	46.0	
	Khosla *et al.* [4]	36.0	
	Song *et al.* [43]	50.0	
	vsLSTM [19]		54.2 ± 0.7
	dppLSTM [19]		54.7 ± 0.7
	SeqDPP [1]		57.4 ± 2.0
	DySeqDPP		$\mathbf{58.4 \pm 2.5}$

Finally, we calculate the precision, recall, and F1-score according to the temporal overlap between the generated summary and the user summaries.

In order to account for the above evaluation scheme, we make some changes to our reinforcement learning algorithm on these two datasets. For training process, firstly we sample the partition proposal l_t with oracle summary based on the old policy on each time step. Thus we can utilize the diagonal values of $\boldsymbol{L}^t$ as shot-level scores and then generate the video summary using the approach described above. Consequently, we can get the reward (F1-score) with the generated video summary. Note that the trajectory $\boldsymbol{\tau}$ here is the oracle summary. Therefore, we can optimize the dynamic SeqDPP with reinforcement learning.

Comparison Results. Table 2 shows the comparison results between our DySeqDPP and several baselines. Note that some of the baseline methods are unsupervised so they are tuned to achieve the best results on the test set. Nonetheless, the supervised ones in general perform better than them. Both SeqDPP and DySeqDPP significantly outperform the others and DySeqDPP ranks to the first by a big margin on SumMe.

Qualitative Results. Figure 4 demonstrates some exemplar video summaries generated by SeqDPP and our DySeqDPP, respectively. It is interesting to see that DySeqDPP captures some shots that are key for the story flow and are yet missed by SeqDPP. Take the first video for instance. The sky diver shows up only at the end of SeqDPP's summary while s/he is kept at both the beginning and the end of DySeqDPP's summary. The second is an amusing video recording how a bird saves a ball from a dog's mouth. However, SeqDPP fails to select the key shot in which the dog bites the ball.

6 Conclusion

In this paper, we study *"how local"* the local diversity should be for video summarization and utilize it as a guideline to devise a sequential model to tackle the dynamic diverse subset selection problem. Furthermore, we apply reinforcement inference [25] in the dynamic seqDPP model to solve the problem of *exposure bias* [33] as well as the issue of non-differentiable metrics existing in SeqDPP [1]. The proposed DySeqDPP can not only seek the appropriate and possibly different lengths of segments dynamically, but also bridge the training and validation phases. Experimental results on video summarization demonstrate the effectiveness of our approach.

Acknowledgements. This work was supported in part by NSF IIS 1741431 & 1566511, gifts from Adobe, and gift GPUs from NVIDIA. B.G. would like to thank Trevor Darrell, Charless Fowlkes, Alexander Ihler, Dequan Wang, and Huazhe Xu for the insightful discussions on SeqDPP which inspired this work.

References

1. Gong, B., Chao, W., Grauman, K., Sha, F.: Diverse sequential subset selection for supervised video summarization. In: Advances in Neural Information Processing Systems (NIPS), pp. 2069–2077 (2014)
2. Money, A.G., Agius, H.: Video summarisation: a conceptual framework and survey of the state of the art. J. Vis. Commun. Image Represent. **19**(2), 121–143 (2008)
3. Hong, R., Tang, J., Tan, H.K., Yan, S., Ngo, C., Chua, T.S.: Event driven summarization for web videos. In: The First SIGMM Workshop on Social Media, pp. 43–48. ACM (2009)
4. Khosla, A., Hamid, R., Lin, C.J., Sundaresan, N.: Large-scale video summarization using web-image priors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2698–2705 (2013)
5. Ngo, C.W., Ma, Y.F., Zhang, H.J.: Automatic video summarization by graph modeling. In: IEEE The Ninth International Conference on Computer Vision (ICCV), pp. 104–109 (2003)
6. Liu, T., Kender, J.R.: Optimization algorithms for the selection of key frame sequences of variable length. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part IV. LNCS, vol. 2353, pp. 403–417. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-47979-1_27
7. Zhang, H.J., Wu, J., Zhong, D., Smoliar, S.W.: An integrated system for content-based video retrieval and browsing. Pattern Recognit. **30**(4), 643–658 (1997)
8. Lee, Y.J., Ghosh, J., Grauman, K.: Discovering important people and objects for egocentric video summarization. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1346–1353. IEEE (2012)

9. Lu, Z., Grauman, K.: Story-driven summarization for egocentric video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2714–2721 (2013)
10. Gygli, M., Grabner, H., Van Gool, L.: Video summarization by learning submodular mixtures of objectives. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3090–3098 (2015)
11. Zhang, K., Chao, W.L., Sha, F., Grauman, K.: Summary transfer: exemplar-based subset selection for video summarization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1059–1067 (2016)
12. Gygli, M., Grabner, H., Riemenschneider, H., Van Gool, L.: Creating summaries from user videos. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part VII. LNCS, vol. 8695, pp. 505–520. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10584-0_33
13. Chao, W., Gong, B., Grauman, K., Sha, F.: Large-margin determinantal point processes. In: Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence (UAI), pp. 191–200 (2015)
14. Kang, H.W., Chen, X.Q.: Space-time video montage. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Volume 2, pp. 1331–1338. IEEE (2006)
15. Ma, Y.F., Lu, L., Zhang, H.J., Li, M.: A user attention model for video summarization. In: The tenth ACM International Conference on Multimedia, pp. 533–542. ACM (2002)
16. Xu, J., Mukherjee, L., Li, Y., Warner, J., Rehg, J.M., Singh, V.: Gaze-enabled egocentric video summarization via constrained submodular maximization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2235–2244 (2015)
17. Sharghi, A., Gong, B., Shah, M.: Query-focused extractive video summarization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part VIII. LNCS, vol. 9912, pp. 3–19. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_1
18. Sharghi, A., Laurel, J.S., Gong, B.: Query-focused video summarization: dataset, evaluation, and a memory network based approach. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2127–2136. IEEE (2017)
19. Zhang, K., Chao, W.-L., Sha, F., Grauman, K.: Video summarization with long short-term memory. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016, Part VII. LNCS, vol. 9911, pp. 766–782. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_47
20. Kulesza, A., Taskar, B.: Learning determinantal point processes. In: Proceedings of 27th Conference on Uncertainty in Artificial Intelligence (UAI), pp. 419–427 (2011)
21. Chao, W.L., Gong, B., Grauman, K., Sha, F.: Large-margin determinantal point processes. In: UAI (2015)
22. Affandi, R.H., Fox, E.B., Adams, R.P., Taskar, B.: Learning the parameters of determinantal point process kernels. In: ICML, pp. 1224–1232 (2014)
23. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Series B **39**, 1–38 (1977)
24. Welch, L.R.: Hidden markov models and the Baum-Welch algorithm. IEEE Inf. Theory Soc. Newsl. **53**(4), 10–13 (2003)
25. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction, vol. 1. MIT press, Cambridge (1998)

26. Kwon, J., Lee, K.M.: A unified framework for event summarization and rare event detection from multiple views. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1737–1750 (2015)
27. Kim, G., Sigal, L., Xing, E.P.: Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4225–4232 (2014)
28. Xiong, B., Grauman, K.: Detecting snap points in egocentric video with a web photo prior. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 282–298. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_19
29. Chu, W.S., Song, Y., Jaimes, A.: Video co-summarization: video summarization by visual co-occurrence. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3584–3592 (2015)
30. Liu, D., Hua, G., Chen, T.: A hierarchical visual model for video object summarization. IEEE Trans. Pattern Anal. Mach. Intell. **32**(12), 2178–2190 (2010)
31. del Molino, A.G., Tan, C., Lim, J.H., Tan, A.H.: Summarization of egocentric videos: a comprehensive survey. IEEE Trans. Hum.-Mach. Syst. **47**(1), 65–76 (2017)
32. Yao, T., Mei, T., Rui, Y.: Highlight detection with pairwise deep ranking for first-person video summarization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 982–990 (2016)
33. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. arXiv preprint arXiv:1511.06732 (2015)
34. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 311–318 (2002)
35. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proceedings of the ACL 2004 Workshop on Text summarization Branches Out, vol. 8, Barcelona, Spain (2004)
36. Banerjee, S., Lavie, A.: Meteor: an automatic metric for MT evaluation with improved correlation with human judgments. In: The ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization. vol. 29, pp. 65–72 (2005)
37. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: consensus-based image description evaluation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4566–4575 (2015)
38. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**(3–4), 229–256 (1992)
39. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. arXiv preprint arXiv:1612.00563 (2016)
40. Kulesza, A., Taskar, B. Determinantal point processes for machine learning. Found. Trends® Mach. Learn. **5**(2–3), 123–286 (2012)
41. Affandi, R.H., Kulesza, A., Fox, E.B.: Markov determinantal point processes. arXiv preprint arXiv:1210.4850 (2012)
42. Sutton, R., Barto, A.: Reinforcement Learning. MIT Press, Cambridge (1998)
43. Song, Y., Vallmitjana, J., Stent, A., Jaimes, A.: TVSUM: summarizing web videos using titles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5179–5187 (2015)
44. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

45. Jia Deng, Wei Dong, R.S.L.J.L.K.L., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proceedings of the IEEE Annual Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
46. Lee, Y.J., Grauman, K.: Predicting important objects for egocentric video summarization. Int. J. Comput. Vis. **114**(1), 38–55 (2015)
47. Sharghi, A., Gong, B., Shah, M.: Query-focused extractive video summarization. In: European Conference on Computer Vision (2016)
48. Szegedy, C., et al.: Going deeper with convolutions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015
49. Potapov, D., Douze, M., Harchaoui, Z., Schmid, C.: Category-specific video summarization. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part VI. LNCS, vol. 8694, pp. 540–555. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_35
50. Li, Y., Merialdo, B.: Multi-video summarization based on video-MMR. In: 2010 11th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), pp. 1–4. IEEE (2010)
51. Zhao, B., Xing, E.P.: Quasi real-time summarization for consumer videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2513–2520 (2014)