Improving Sequential Determinantal Point Processes for Supervised Video Summarization: Supplementary Material

Aidean Sharghi[1 [0000000320051334]], Ali Borji[1 [0000000181980335]], Chengtao Li[2 [0000000323462753]], Tianbao Yang[3 [0000000278585438]], and Boqing Gong[4 [0000000339155977]]

[1] Center for Research in Computer Vision, University of Central Florida, Orlando, Florida
[2] Massachusetts Institute of Technology, Cambridge, Massachusetts
[3] University of Iowa, Iowa City, Iowa
[4] Tencent AI Lab, Seattle, Washington

Here we provide the supplementary materials to support the main text.

Section A derives the normalization constant for *G*DPPs.
Section B describes two approaches to sample from *G*DPP distribution.
Section C describes the algorithm to aggregate user summaries into an *oracle* summary.
Section D compares our *large-margin* and Seq*G*DPP models to state-of-the-art query-focused video summarization frameworks.

A Normalization of GDPPs

To compute the normalization constant Z_G for GDPP, we have to sum over all the possible subsets of the ground set $\boldsymbol{y} \subseteq \mathcal{Y}$:

$$\begin{aligned} Z_G &\triangleq \sum_{\boldsymbol{y}} \sum_{k=0}^{\mathsf{N}} \pi(\kappa = k) \sum_{J \subseteq \mathcal{Y}, |J|=k} P(Y = \boldsymbol{y}; V_J) \prod_{n \in J} \lambda_n \\ &= \sum_{k=0}^{\mathsf{N}} \pi(\kappa = k) \sum_{|J|=k} \sum_{\boldsymbol{y}} P(Y = \boldsymbol{y}; V_J) \prod_{n \in J} \lambda_n \\ &= \sum_{k=0}^{\mathsf{N}} \pi(\kappa = k) \sum_{|J|=k} \prod_{n \in J} \lambda_n \triangleq \sum_{k=0}^{\mathsf{N}} \pi(k)\, e_{\mathsf{N}}(k), \end{aligned} \tag{1}$$

where $\pi(k) \triangleq \pi(\kappa = k)$, and $e_{\mathsf{N}}(k) \triangleq \sum_{|J|=k} \prod_{n \in J} \lambda_n$ is the elementary symmetric polynomial over the eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{\mathsf{N}}$ of the DPP kernel $\boldsymbol{L}$.

Therefore, the computational complexity of the normalization constant Z_G for GDPP depends on the choice of the "size" distribution $\pi(k)$ and the complexity to compute $e_{\mathsf{N}}(k)$. Evaluating $\pi(k)$ usually takes less than $O(\mathsf{N}^2)$ time for $\pi(\kappa)$ being any popular discrete distribution. By Newton's identities or the recursive algorithm [6, Algorithm 7] we can compute all the elementary symmetric polynomials $e_{\mathsf{N}}(k), k = 1, 2, \cdots \mathsf{N}$, in $O(\mathsf{N}^2)$ time. Overall, the normalization complexity of GDPP hinges on the eigen-decomposition of $\boldsymbol{L}$ for obtaining λ_n in eq. (1), and is about the same as the complexity of normalizing an L-ensemble DPP (i.e., computing $\det(\boldsymbol{L} + \boldsymbol{I})$).

B Sampling from GDPP

We consider exact sampling method assuming an already existing eigendecomposition of L and a Markov chain for sampling GDPP.

B.1 Sampling via Eigendecomposition

Since GDPP could be viewed as a weighted mixture of k-DPP, we have the following decomposition of the probability:

$$P(S|S \sim \text{GDPP}) = P(S|S \sim k\text{-DPP})P(k|k \sim \text{GDPP}) \tag{2}$$

Here with a slight abuse of notation we let $k \sim$GDPP to denote the probability of sampling a k-DPP from GDPP. With aforementioned properties we have

$$P(k|k \sim \text{GDPP}) \propto \pi(k)e_k(L) \tag{3}$$

which can be computed in $\mathcal{O}(Nk + k^2)$ if we already have access to eigenvalues of L.

Hence, we can employ a 2-phase procedure of first deciding which k-DPP to sample from, and then sampling from corresponding k-DPP. The procedure is shown in Algorithm 1.

Algorithm 1 Sampling from GDPP via Eigendecomposition

Require: L the GDPP kernel, π the (unnormalized) size probability, $k^{\max}$ and $k^{\min}$ as stated in the assumption, V the ground set
Ensure: S sampled from GDPP
Sample k from discrete distribution $\{p_i = \pi(i)e_i(L)\}_{i=0}^{N}$
Sample S from k-DPP

B.2 Sampling via Markov Chain

We consider sampling from GDPP via Markov chains. While running the chain, we maintain a currently active set as the current state for the chain, and in each iteration we try to update the current active set only slightly with certain probabilities such that the update is efficient to be done and the stable distribution – the distribution of active set when running chain long enough – is the same as GDPP. Since in each iteration we only update the active set slightly, the probability of transferring from one state to any other states in the chain is non-zero. To fulfill this condition, it suffices to assume that $\pi_i > 0$ for all i such that, there exists $j < i$ and $k > i$ such that $\pi_j > 0$ and $\pi_k > 0$.

The sampling efficiency of Markov chain heavily depends on its mixing time. Thus, besides constructing the chain, we also show that the constructed chains are fast mixing.

We construct an add-delete Markov chain to sample from GDPP. Concretely, we update the active set by either trying to add an element to or delete an element from it with certain transition probabilities. The full algorithm is shown in Algorithm 2.

Algorithm 2 Add-Delete Markov Chain for GDPP

Require: L the GDPP kernel, π the (unnormalized) size probability, $k^{\max}$ and $k^{\min}$ as stated in the assumption, V the ground set
Ensure: S sampled from GDPP
  Initialize S s.t. $P(S) > 0$
  while not mixed **do**
    Let $b = 1$ with probability $\frac{1}{2}$
    if $b = 1$ **then**
      Pick $s \in V$ uniformly randomly
      if $s \notin S$ and $|S| < k^{\max}$ **then**
        $S \leftarrow S \cup \{s\}$ with probability $p^+(S, s) = \frac{F(S \cup \{s\})}{F(S)+F(S \cup \{s\})}$
      else if $s \in S$ and $|S| > k^{\min}$ **then**
        $S \leftarrow S \backslash \{s\}$ with probability $p^-(S, s) = \frac{F(S \backslash \{s\})}{F(S)+F(S \backslash \{s\})}$
      end if
    else
      Do nothing
    end if
  end while

Mixing Time To show fast mixing, we consider using *path coupling*, which essentially says that if we have a contraction of two (coupling) chains then we have fast mixing. Assume we have a chain (S_t) on state space 2^V with transition matrix P, a *coupling* is a new chain (S_t, Y_t) on $V \times V$ such that both (S_t) and (Y_t), if considered marginally, are Markov chains with the same transition matrices P. The key point of coupling is to construct such a new chain to encourage S_t and Y_t to *coalesce* quickly. If, in the new chain, $\Pr(S_t \neq Y_t) \leq \varepsilon$ for some fixed t regardless of the starting state (S_0, Y_0), then $\tau(\varepsilon) \leq t$ [1]. To make the coupling construction easier, *Path coupling* [3] is then introduced so as to reduce the coupling to adjacent states in an appropriately constructed state graph. The coupling of arbitrary states follows by aggregation over a path between the two. Path coupling is formalized in the following lemma.

Lemma 1. *[3,4] Let δ be an integer-valued metric on $V \times V$ where $\delta(\cdot, \cdot) \leq D$. Let E be a subset of $V \times V$ such that for all $(S_t, Y_t) \in V \times V$ there exists a path $S_t = Z^0, \ldots, Z^r = Y_t$ between S_t and Y_t where $(Z^i, Z^{i+1}) \in E$ for $i \in [r - 1]$ and $\sum_i \delta(Z^i, Z^{i+1}) = \delta(S_t, Y_t)$. Suppose a coupling $(S, T) \rightarrow (S', T')$ of the Markov chain is defined on all pairs in E such that there exists an $\alpha < 1$ such that $\mathbb{E}[\delta(S', T')] \leq \alpha\delta(S, T)$ for all $(S, T) \in E$, then we have $\tau(\varepsilon) \leq \frac{\log(D\varepsilon^{-1})}{(1-\alpha)}$.*

With path coupling we are able to bound the mixing time of Algo. 2 as follows.

Theorem 1 *Let* $\alpha = \max_{(S,T)\in E}\{\alpha_1, \alpha_2\}$ *where* α_1 *and* α_2 *are defined as* $[\![$

$$\begin{aligned}\alpha_1 =& 1 - [\![|T| > k^{\min}]\!] \sum_{i\in T} |p^-(T,i) - p^-(S,i)|_+ \\ & - [\![|S| < k^{\max}]\!] \sum_{i\in V\setminus S} |p^+(S,i) - p^+(T,i)|_+; \\ \alpha_2 =& [\![|S| > k^{\min}]\!](\min\{p^-(S,s), p^-(T,t)\} - \sum_{i\in R} |p^-(S,i) - p^-(T,i)|) + \\ & [\![|S| < k^{\max}]\!](\min\{p^+(S,t), p^+(T,s)\} - \sum_{i\in V\setminus(S\cup T)} |p^+(S,i) - p^+(T,i)|).\end{aligned}$$

In the expression, summations over absolute difference quantifies the sensitivity of transition probabilities to adding/deleting elements in neighboring (S,T) *in* E*. Assuming* $\alpha < 1$*, we have*

$$\tau(\varepsilon) \le \frac{2N\log(k^{\max}\varepsilon^{-1})}{1-\alpha}$$

Proof. We define $\delta(X,Y) = \frac{1}{2}(|X \oplus Y| + ||X| - |Y||)$. It is clear that $\delta(X,Y) \ge 1$ for $X \ne Y$. Let $E = \{(X,Y) : \delta(X,Y) = 1\}$ be the set of adjacent states (neighbors), and it follows that $\delta(\cdot,\cdot)$ is a metric satisfying conditions in Lemma 1. Also we have $\delta(X,Y) \le k^{\max}$.

We consider constructing a path coupling between any two states S and T with $\delta(S,T) = 1$, S' and T' be the two states after transition. We sample $c_S, c_T \in \{0,1\}$, if c_S is 0 then $S' = S$ and the same with c_T. $i_S, i_T \in V$ are drawn uniformly randomly. We consider two possible settings for S and T:

1. If S or T is a subset of the other, we assume without of generality that $S = T \cup \{t\}$. In this setting we always let $i_S = i_T = i$. Then
   (a) If $i = t$, we let $c_S = 1 - c_T$;
      i. If $c_S = 1$ then $\delta(S',T') = 0$ with probability $p^-(S,t)$;
      ii. If $c_S = 0$ then $\delta(S',T') = 0$ with probability $p^+(T,t)$;
   (b) If $i \in T$, we set $c_S = c_T$;
      i. If $c_S = 1$ and $|T| > k^{\min}$ then $\delta(S',T') = 2$ with probability $(p^-(T,i) - p^-(S,i))_+$;
   (c) If $i \in V\setminus S$, we set $c_S = c_T$;
      i. If $c_S = 1$ and $|S| < k^{\max}$ then $\delta(S',T') = 2$ with probability $(p^+(S,i) - p^+(T,i))_+$.
2. If S and T are of the same sizes, let $S = R \cup \{s\}$ and $T = R \cup \{t\}$. In this setting we always let $c_S = c_T = c$. We consider the case of $c = 1$:
   (a) If $i_S = s$, let $i_T = t$. If $|S| > k^{\min}$, then $\delta(S',T') = 0$ with probability $\min\{p^-(S,s), p^-(T,t)\}$;
   (b) If $i_S = t$, let $i_T = s$. If $|S| < k^{\max}$, then $\delta(S',T') = 0$ with probability $\min\{p^+(S,t), p^+(T,s)\}$;
   (c) If $i_S \in R$, let $i_T = i_S$. If $|S| > k^{\min}$, then $\delta(S',T') = 2$ with probability $|p^-(S,i_S) - p^-(T,i_T)|$;

(d) If $i_S \in V \backslash (S \cup T)$, let $i_T = i_S$. If $|S| < k^{\max}$, then $\delta(S', T') = 2$ with probability $|p^+(S, i_S) - p^+(T, i_T)|$.

In all cases where we didn't specify $\delta(S', T')$, it will be $\delta(S', T') = 1$. In the first case of $S = T \cup \{t\}$ we have

$$\begin{aligned}\frac{\mathbb{E}[\delta(S', T')]}{\mathbb{E}[\delta(S, T)]} \leq & \frac{1}{2N}((1 - p^-(S, t)) + (1 - p^+(T, t)) + \\ & (2|T| + [\![|T| > k^{\min}]\!] \sum_{i \in T} (p^-(T, i) - p^-(S, i))_+) + \\ & (2(N - |S|) + [\![|S| < k^{\max}]\!] \sum_{i \in V \backslash S} (p^+(S, i) - p^+(T, i))_+)) \\ = 1 - \frac{1}{2N}(1 - & [\![|T| > k^{\min}]\!] \sum_{i \in T} (p^-(T, i) - p^-(S, i))_+ \\ & - [\![|S| < k^{\max}]\!] \sum_{i \in V \backslash S} (p^+(S, i) - p^+(T, i))_+) = 1 - \frac{\alpha_1}{2N},\end{aligned}$$

while in the second case of $|S| = R \cup \{s\}$ and $T = R \cup \{t\}$ we have

$$\begin{aligned}\frac{\mathbb{E}[\delta(S', T')]}{\mathbb{E}[\delta(S, T)]} \leq & \frac{1}{2N}((1 - [\![|S| > k^{\min}]\!] \min\{p^-(S, s), p^-(T, t)\}) + \\ & (1 - [\![|S| < k^{\max}]\!] \min\{p^+(S, t), p^+(T, s)\}) + \\ & (2|R| + [\![|S| > k^{\min}]\!] \sum_{i \in R} |p^-(S, i) - p^-(T, i)|) + \\ & (2(N - |S| - 1) + [\![|S| < k^{\max}]\!] \sum_{i \in V \backslash (S \cup T)} |p^+(S, i) - p^+(T, i)|)) \\ = 1 - \frac{1}{2N}([\![|S| > k^{\min}]\!] & \min\{p^-(S, s), p^-(T, t)\} - \sum_{i \in R} |p^-(S, i) - p^-(T, i)| + \\ & [\![|S| < k^{\max}]\!](\min\{p^+(S, t), p^+(T, s)\} - \\ & \sum_{i \in V \backslash (S \cup T)} |p^+(S, i) - p^+(T, i)|)) = 1 - \frac{\alpha_2}{2N}.\end{aligned}$$

Let $\alpha = \max_{(S,T) \in E}\{\alpha_1, \alpha_2\}$. If $\alpha < 1$, with Lemma 1 we have

$$\tau(\varepsilon) \leq \frac{2N \log(k^{\max}/\varepsilon)}{1 - \alpha}.$$

Remarks This bound involves some constants that is hard to compute in practice. It is more general due to the generality of GDPP – when setting π to be uniform in $[k]$ and set $k^{\min} = 0$, $k^{\max} = k$, we recover the bound in [9].

C Constructing Oracle Summaries

Supervised video summarization approaches are conventionally trained on one target summary per video, i.e. *oracle summary*. In our annotation collection stage, we ob-

Table 1: Comparison results for query-focused video summarization (%). AUCs are computed on the curves drawn in Figure(1) until the 60 seconds mark. Matching colors indicate the base-model and its equivalent large-margin peer.

	AUC_{Π}	$\text{AUC}_{\text{Gaussian}}$
SH-DPP [10]	7.03	6.92
LSTM-DPP [13]	5.99	5.97
SeqDPP [5]	6.14	6.18
LM-SeqDPP	9.70	9.66
MemNet [11]	9.93	9.77
LM-MemNet	9.83	9.67
SeqGDPP	9.72	9.66
LM-SeqGDPP	10.43	**10.30**

tained 3 user summaries per video, however, to train the models, we aggregate them into one *oracle summary* using a greedy algorithm that has been used in several previous works [5,10,11]. To construct an oracle summary, the algorithm [7] begins with the set of common shots (y^0) of all three human-generated summaries and at each iteration, it appends to y^0 the shot that results in the largest marginal gain in $G(i)$,

$$G(i) = \sum_u \text{F1}(y^0 \cup i, y_u) - \sum_u \text{F1}(y^0, y_u) \tag{4}$$

where u iterates over three human-generated summaries and F1 is the harmonic F-score obtained using the evaluation metric.

D Query-Focused Video Summarization

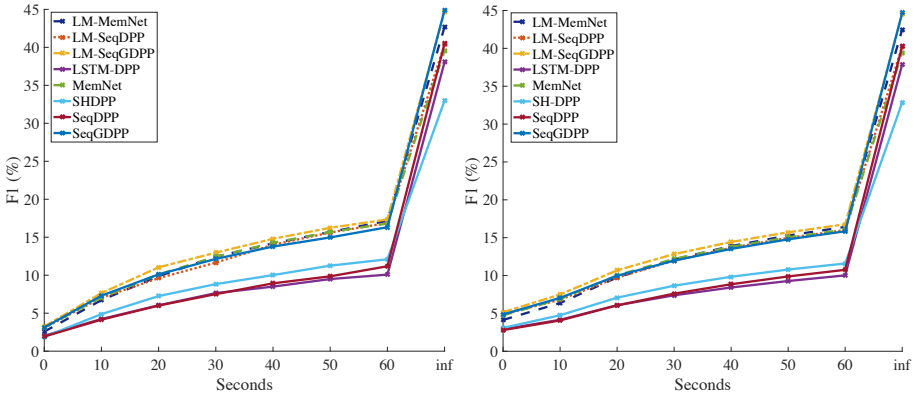
In [10], Sharghi et al. first defined query-oriented video summarization, that is given a $\{$*video*,*query*$\}$ pair, system summary must include both contextually important segments and preferences (query) indicated by the user at the same time. In this section, we assess the performance of our models and compare them to several baselines.

Dataset. [11] compiled a dataset tailored specifically for query-focused video summarization, built upon UT Egocentric videos [8]. For each of the four videos in UTE, 46 queries are defined each as a pair of visual concepts, and for every $\{$*video*,*query*$\}$ pair, 3 human-generated summaries are collected.

Features. We follow [10,11] in extracting the 76-dimensional concept detection scores obtained by running classifiers from SentiBank [2] as feature representation.

Baselines. We compare our models to state-of-the-art query-focused summarization algorithms as well as other baselines including:

– *Sequential-Hierarchical DPP [10].* SH-DPP consists of two stacked SeqDPP layers where the top layer summarizes the video by only adding query-relevant shots to



(a) Π temporal filter **(b)** Gaussian temporal filter

Fig. 1: Comparison results for query-focused video summarization task. x axis represent the temporal filter parameter. In case of Π filter, it indicates how far a match can be temporally (in terms of seconds), whereas in the Gaussian filter, it is the kernel bandwidth.

the summary, and the bottom layer completes the summary by adding to it a diverse set of contextually important events that might not reflect the queries but are necessary to make a coherent story. Furthermore, the bottom layer is conditioned on the output of top layer, hence, minimizing the redundancy of information and maximizing diversity between the shots included by two layers.

– *LSTM-DPP [13].* To make this framework compatible for query-focused summarization, we concatenate the features with the query and feed them to the network.

– *MemNet [11].* This framework takes advantage of strong modeling capacity in memory networks [12] to generate query-dependent features jointly from static visual features and the query. These features are then fed to a SeqDPP layer for summarization.

In Figure(1) and Table(1), we compare the baselines with Seq*G*DPP and large-margin modification of SeqDPP and MemNet. Comparing base SeqDPP with LM-SeqDPP, the performance gain is quite significant. On the other hand, MemNet preforms competitive to LM-MemNet. LM-Seq*G*DPP outperforms all the models, and this is in line with our generic video summarization results reported in Table(2) of the main text.

References

1. Aldous, D.J.: Some inequalities for reversible markov chains. Journal of the London Mathematical Society pp. 564–576 (1982) 3
2. Borth, D., Chen, T., Ji, R., Chang, S.F.: Sentibank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content. In: Proceedings of the 21st ACM international conference on Multimedia. pp. 459–460. ACM (2013) 6
3. Bubley, R., Dyer, M.: Path coupling: A technique for proving rapid mixing in markov chains. In: focs. pp. 223–231 (1997) 3

4. Dyer, M., Greenhill, C.: A more rapidly mixing markov chain for graph colorings. Random Structures and Algorithms pp. 285–317 (1998) 3
5. Gong, B., Chao, W.L., Grauman, K., Sha, F.: Diverse sequential subset selection for supervised video summarization. In: Advances in Neural Information Processing Systems. pp. 2069–2077 (2014) 6
6. Kulesza, A., Taskar, B.: Determinantal point processes for machine learning. Foundations and Trends® in Machine Learning **5**(2–3), 123–286 (2012) 1
7. Kulesza, A., Taskar, B., et al.: Determinantal point processes for machine learning. Foundations and Trends® in Machine Learning **5**(2–3), 123–286 (2012) 6
8. Lee, Y.J., Ghosh, J., Grauman, K.: Discovering important people and objects for egocentric video summarization. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 1346–1353. IEEE (2012) 6
9. Li, C., Jegelka, S., Sra, S.: Fast sampling for strongly rayleigh measures with application to determinantal point processes. arXiv preprint arXiv:1607.03559 (2016) 5
10. Sharghi, A., Gong, B., Shah, M.: Query-focused extractive video summarization. In: European Conference on Computer Vision. pp. 3–19. Springer (2016) 6
11. Sharghi, A., Laurel, J.S., Gong, B.: Query-focused video summarization: Dataset, evaluation, and a memory network based approach. arXiv preprint arXiv:1707.04960 (2017) 6, 7
12. Sukhbaatar, S., Weston, J., Fergus, R., et al.: End-to-end memory networks. In: Advances in neural information processing systems. pp. 2440–2448 (2015) 7
13. Zhang, K., Chao, W.L., Sha, F., Grauman, K.: Video summarization with long short-term memory. In: European Conference on Computer Vision. pp. 766–782. Springer (2016) 6, 7