Unsupervised Video Summarization with Adversarial LSTM Networks

Behrooz Mahasseni, Michael Lam and Sinisa Todorovic
Oregon State University
Corvallis, OR
behrooz.mahasseni@gmail.com {lamm,sinisa}@oregonstate.edu

Abstract

This paper addresses the problem of unsupervised video summarization, formulated as selecting a sparse subset of video frames that optimally represent the input video. Our key idea is to learn a deep summarizer network to minimize distance between training videos and a distribution of their summarizations, in an unsupervised way. Such a summarizer can then be applied on a new video for estimating its optimal summarization. For learning, we specify a novel generative adversarial framework, consisting of the summarizer and discriminator. The summarizer is the autoencoder long short-term memory network (LSTM) aimed at, first, selecting video frames, and then decoding the obtained summarization for reconstructing the input video. The discriminator is another LSTM aimed at distinguishing between the original video and its reconstruction from the summarizer. The summarizer LSTM is cast as an adversary of the discriminator, i.e., trained so as to maximally confuse the discriminator. This learning is also regularized for sparsity. Evaluation on four benchmark datasets, consisting of videos showing diverse events in first- and third-person views, demonstrates our competitive performance in comparison to fully supervised state-of-the-art approaches.

1. Introduction

A wide range of applications require automated summarization of videos [36, 42], e.g., for saving time of human inspection, or enabling subsequent video analysis. Depending on the application, there are various distinct definitions of video summarization [30, 27, 28, 1, 40, 37, 6, 20, 18, 14, 25, 12]. In this paper, we consider unsupervised video summarization, and cast it as a key frame selection problem. Given a sequence of video frames, our goal is to select a sparse subset of frames such that a representation error between the video and its summary is minimal.

Our problem statement differs from other formulations considered in the literature, for example, when a particular domain of videos to be summarized is a priori known (e.g., first-person videos) [18], or when ground-truth annotations of key frames are provided in training data based on attention, aesthetics, quality, landmark presence, and certain object occurrences and motions [9].

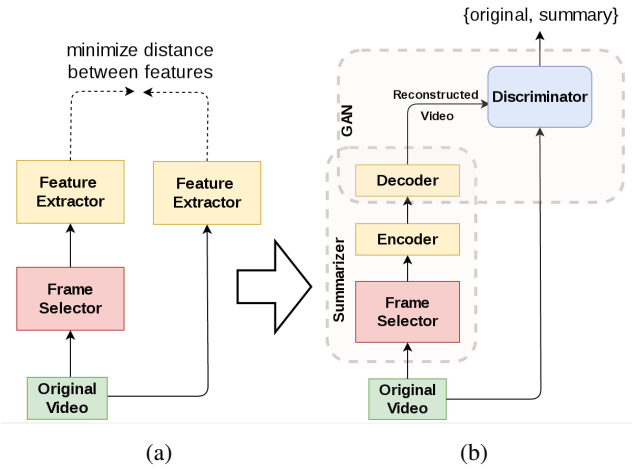


Figure 1: (a) Overview: Our goal is to select key frames such that a distance between feature representations of the selected key frames and the video is minimized. (b) As specifying a suitable distance between deep features is difficult, we use a generative adversarial framework for optimizing the frame selector. Our approach consists of a variational auto-encoder and a generative adversarial network.

Fig. 1a shows an overview of our approach to selecting key frames from a given video. The key frame selector is learned so as to minimize a distance between features extracted from the video and the selected key frames. Following recent advances in deep learning [35, 41, 43], we extract deep features from both the video and selected sequence of key frames using a cascade of a Convolutional Neural Network (CNN) – specifically GoogleNet [38] – and Long Short-Term Memory Network (LSTM) [13, 35]. The CNN is grounded onto pixels and extracts deep features from a given frame. The LSTM then fuses a sequence of the CNN's

outputs for capturing long-range dependencies among the frames, and produces its own deep feature representing the input sequence. Specifically, we use the (variational) auto-encoder LSTM [35, 16] as a suitable deep architecture for unsupervised learning of video features. Given a distance between the deep representations of the video and selected key frames, our goal is to optimize the frame selector such that this distance is minimized over training examples.

Recent work, however, demonstrates that specifying a suitable distance of deep features is difficult [19]. Hence, we resort to the generative adversarial framework [8], which extends the aforementioned video summarization network with an additional discriminator network. As shown in Fig. 1b, the decoder part of the summarizer is used to reconstruct a video from the sequence of selected key frames. Then, we use a discriminator, which is another LSTM, to distinguish between the original video and its reconstruction from the summarizer. The auto-encoder LSTM and the frame selector are jointly trained so as to maximally confuse the discriminator LSTM – i.e., they are cast in a role of the discriminator's adversary – such that the discriminator has a high error rate in recognizing between the original and reconstructed videos. When this recognition error becomes maximum, we deem that the frame selector is learned to produce optimal video summarizations.

As we will show in this paper, our approach allows for an effective regularization of generative-adversarial learning in terms of: (i) limiting the total number of key frames that can be selected; or (ii) maximizing visual diversity among the selected key frames. For a fair comparison with related approaches to fully supervised video summarization – a different setting from ours that provides access to ground-truth key frame annotations in training – we also show how to effectively incorporate the available supervision as an additional type of regularization in learning.

Evaluation on four benchmark datasets, consisting of videos showing diverse events in first- and third-person views, demonstrates our competitive performance in comparison to fully supervised state-of-the-art approaches.

Our contributions include:

1. A new approach to unsupervised video summarization that combines variational auto-encoders and generative-adversarial training of deep architectures.
2. First specification of generative-adversarial training on high resolution video sequences.

In the following, Sec. 2 reviews prior work, Sec. 3 briefly introduces the generative adversarial network (GAN) and the variational autoencoder (VAE) models, Sec. 4 specifies main components of our approach, Sec. 5 formulates our end-to-end training, Sec. 6 describes variants of our approach differing in types of regularization we use in learning, and finally Sec. 7 presents our results.

2. Related Work

This section reviews related: (i) problem formulations of video summarization; (ii) approaches to supervised and unsupervised video summarization; (iii) deep learning approaches; and (iv) work using the generative adversarial framework in learning.

Various Problem Formulations. Video summarization is a long-standing problem, with various formulations considered in the literature. For example, the video synopsis [28] tracks moving objects, and then packs the identified video tubes into a smaller space-time volume. Also, montages [1, 40, 37] merge and overlaps key frames into a single summary image. Both of these problem formulations, however, do not require that the video summary preserves the information about a temporal layout of motions in the video. Previous work has also studied hyperlapses where the camera viewpoint is being changed during the time-lapse for speeding-up or slowing-down certain parts of the input video [18, 14, 25, 12]. Our problem statement is closest to storyboards, representing a subset of representative video frames [6, 20]. However, except for [43, 41], existing approaches to generating storyboards do not take advantage of deep learning.

Supervised vs. Unsupervised Summarization. Supervised methods assume access to human annotations of key frames in training videos, and seek to optimize their frame selectors so as to minimize loss with respect to this ground truth [7, 43, 42]. However, for a wide range of domains, it may be impossible to provide reliable and a sufficiently large amount of human annotations (e.g., military, nursing homes). These domains have been addressed with unsupervised methods, which typically use heuristic criteria for ranking and selecting key frames [21, 41, 15, 44, 34]. There have been attempts to use transfer learning for domains without supervision [43], but the surprisingly better performance of the transfer learning setting compared to the canonical setting, reported in [43], suggests a high correlation of the domains for three training dataset and one test dataset , which is hard to ensure in real-world settings.

Deep Architectures for Video Summarization. In [43], two LSTMs are used – one along the time sequence and the other in reverse from the video's end – to select key video frames, and trained by minimizing the cross-entropy loss on annotated ground-truth key frames with an additional objective based on determinantal point process (DPP) to ensure diversity of the selected frames. Our main differences are that we do not consider the key frame annotations, and train our LSTMs using the unsupervised generative-adversarial learning. In [41], recurrent auto-encoders are learned to represent annotated temporal intervals in training videos, called highlights. In contrast, we do not require human annotations of highlights in training, and we do not perform temporal video segmentation (highlight vs

non-highlight), but key frame selection.

Generative Adversarial Networks (GANs) have been used for image-understanding problems [8, 29, 33, 31], and frame prediction/generation [22, 39, 5]. But we are not aware of their previous use for video summarization. In [19], the discriminator output of a GAN is used to provide a learning signal for the variational auto-encoder (VAE). We extend this approach in three critical ways: (1) We specify a new variational auto-encoder LSTM, whereas their auto-encoder is not a recurrent neural network, and thus cannot be used for videos; (2) Our generative-adversarial learning additionally takes into account the frame selector – a component not considered in [19]; and (3) We formulate regularization of generative-adversarial learning that is suitable for video summarization.

3. Review of VAE and GAN

Variational Autoencoder (VAE) [16] is a directed graphical model which defines a posterior distribution over the observed data, given an unobserved latent variable. Let $\boldsymbol{e} \sim p_{\boldsymbol{e}}(\boldsymbol{e})$ be a prior over the unobserved latent variable, and $\boldsymbol{x}$ be the observed data. One can interpret $\boldsymbol{e}$ as the encoding of $\boldsymbol{x}$ and define $q(\boldsymbol{e}|x)$ as the probability of observing $\boldsymbol{e}$ given $\boldsymbol{x}$. It is typical to set $p_{\boldsymbol{e}}(\boldsymbol{e})$ as the standard normal distribution. Similarly, $p(\boldsymbol{x}|\boldsymbol{e})$ identifies the conditional generative distribution for $\boldsymbol{x}$. Learning is done by minimizing the negative log-likelihood of the data distribution:

$$-\log \frac{p(\boldsymbol{x}|\boldsymbol{e})p(\boldsymbol{e})}{q(\boldsymbol{e}|\boldsymbol{x})} = \underbrace{-\log(p(\boldsymbol{x}|\boldsymbol{e}))}_{\mathcal{L}_{\text{reconst}}} + \underbrace{D_{KL}(q(\boldsymbol{e}|\boldsymbol{x})\|p(\boldsymbol{e})}_{\mathcal{L}_{\text{prior}}}. \tag{1}$$

For efficient learning, Kingma et al. [16] propose a reparameterization of the variational lower bound suitable for stochastic gradient descent.

Generative Adversarial Network (GAN) [8] is a neural network that consists of two competing subnetworks: i) a 'generator' network (G) which generates data mimicking an unknown distribution and ii) a 'discriminator' network (D) which discriminates between the generated samples and the ones from true observations. The goal is to find a generator which fits the true data distribution while maximizing the probability of the discriminator making a mistake.

Let $\boldsymbol{x}$ be the true data sample, $\boldsymbol{e} \sim p_{\boldsymbol{e}}(\boldsymbol{e})$ be the prior input noise, and $\hat{\boldsymbol{x}} = G(\boldsymbol{e})$ be the generated sample. Learning is formulated as the following minimax optimization:

$$\min_G \max_D \big[\underbrace{\mathbb{E}_{\boldsymbol{x}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{e}}[\log(1 - D(\hat{\boldsymbol{x}}))]}_{\mathcal{L}_{\text{GAN}}}\big], \tag{2}$$

where D is trained to maximize the probability of correct sample classification (true vs generated) and G is simultaneously trained to minimize $\log(1 - D(\hat{\boldsymbol{x}}))$.

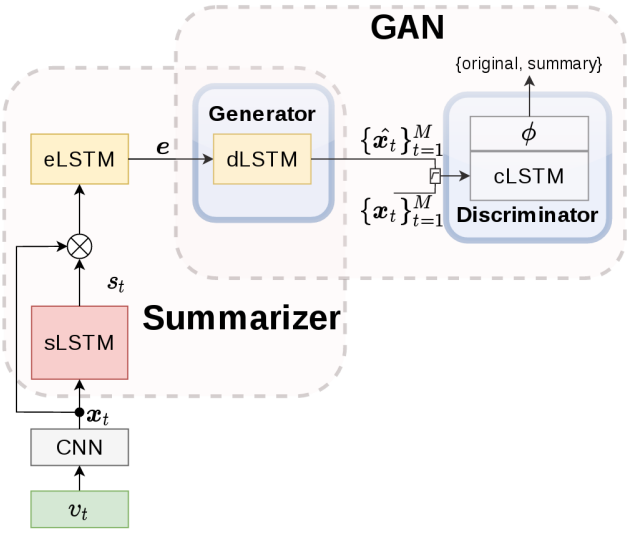


Figure 2: Main components of our approach: The selector LSTM (sLSTM) selects a subset of frames from the input sequence $\boldsymbol{x}$. The encoder LSTM (eLSTM) encodes the selected frames to a fixed-length feature $\boldsymbol{e}$, which is then forwarded to the decoder LSTM (dLSTM) for reconstructing a video $\hat{\boldsymbol{x}}$. The discriminator LSTM (cLSTM) classifies $\hat{\boldsymbol{x}}$ as 'original' or 'summary' class. dLSTM and cLSTM form the generative adversarial network (GAN).

4. Main Components of Our Approach

Our approach consists of the summarizer and discriminator recurrent networks, as illustrated in Figure 2.

Given CNN's deep features for every frame of the input video, $\boldsymbol{x} = \{\boldsymbol{x}_t : t = 1, \ldots, M\}$, the summarizer uses a selector LSTM (sLSTM) to select a subset of these frames, and then an encoder LSTM (eLSTM) to encode the sequence of selected frames to a deep feature, $\boldsymbol{e}$. Specifically, for every frame $\boldsymbol{x}_t$, sLSTM outputs normalized importance scores $\boldsymbol{s} = \{s_t : s_t \in [0, 1], t = 1, \ldots, M\}$ for selecting the frame. The input sequence of frame features $\boldsymbol{x}$ is weighted with these importance scores, and then forwarded to eLSTM. Note that in the special case of discretized scores, $s_t \in \{0, 1\}$, eLSTM receives only a subset of frames for which $s_t = 1$. The last component of the summarizer is a decoder LSTM (dLSTM), which takes $\boldsymbol{e}$ as input, and reconstructs a sequence of features corresponding to the input video, $\hat{\boldsymbol{x}} = \{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, ..., \hat{\boldsymbol{x}}_M\}$.

The discriminator is aimed at distinguishing between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ as belonging to two distinct classes: 'original' and 'summary'. This classifier can be viewed as estimating a distance between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, and assigning distinct class la-

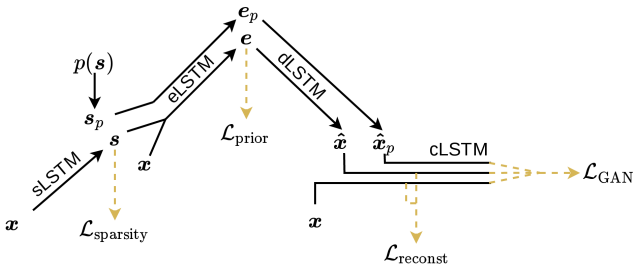


Figure 3: The four loss functions used in our training. $\mathcal{L}_{\text{GAN}}$ is the augmented GAN loss and $\mathcal{L}_{\text{reconst}}$ is the reconstruction loss for the recurrent encoder-decoder. In training, we use an additional frame selector $\boldsymbol{s}_p$, governed by a prior distribution (e.g., uniform), which produces the encoded representation $\boldsymbol{e}_p$, and the reconstructed feature sequence $\hat{\boldsymbol{x}}_p$. The adversarial training of cLSTM is regularized such that it is highly accurate on recognizing $\hat{\boldsymbol{x}}_p$ as 'summary', but that it confuses $\hat{\boldsymbol{x}}$ as 'original'.

bels to $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ if their distance is sufficiently large. In this sense, the discriminator serves to estimate a representation error between the original video and our video summarization. While one way to implement the discriminator could be an energy-based encoder-decoder [45], in our experiments a binary sequence classifier have shown better performance. Hence, we specify the discriminator as a classifier LSTM (cLSTM) with a binary-classification output.

Analogous to the generative adversarial networks presented in [8, 19], we have that dLSTM and cLSTM form the generative adversarial network (GAN). The summarizer and discriminator networks are trained adversarially until the discriminator is not able to discriminate between the reconstructed videos from summaries and the original videos.

5. Training of sLSTM, eLSTM, and dLSTM

This section specifies our learning of: (i) Summarizer parameters, $\{\theta_s, \theta_e, \theta_d\}$, characterizing sLSTM, eLSTM, and dLSTM; and (ii) GAN parameters, $\{\theta_d, \theta_c\}$, defining dLSTM and cLSTM. Note that θ_d are shared parameters between the summarizer and GAN.

As illustrated in Fig. 3, our training is defined by four loss functions: 1) Loss of GAN, $\mathcal{L}_{\text{GAN}}$, 2) Reconstruction loss for the recurrent encoder-decoder, $\mathcal{L}_{\text{reconst}}$, 3) Prior loss, $\mathcal{L}_{\text{prior}}$, and 4) Regularization loss, $\mathcal{L}_{\text{sparsity}}$. The key idea behind our generative-adversarial training is to introduce an additional frame selector $\boldsymbol{s}_p$, governed by a prior distribution (e.g., uniform distribution), $\boldsymbol{s}_p \sim p(\boldsymbol{s}_p)$. Sampling the input video frames with $\boldsymbol{s}_p$ gives a subset which is passed to eLSTM, producing the encoded representation $\boldsymbol{e}_p$. Given $\boldsymbol{e}_p$, dLSTM reconstructs a video sequence $\hat{\boldsymbol{x}}_p$. We use $\hat{\boldsymbol{x}}_p$ to regularize learning of the discriminator, such that cLSTM is highly accurate on recognizing $\hat{\boldsymbol{x}}_p$ as the 'summary' class, but that it confuses $\hat{\boldsymbol{x}}$ as 'original' class. Recall that the $\mathcal{L}_{\text{prior}}$ is imposed by the prior distribution over $\boldsymbol{e}$ as in (1).

Similar to the training of GAN models in [8, 19], we formulate an adversarial learning algorithm that iteratively optimizes the following three objectives:

1. For learning $\{\theta_s, \theta_e\}$, minimize $(\mathcal{L}_{\text{reconst}} + \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{sparsity}})$.
2. For learning θ_d, minimize $(\mathcal{L}_{\text{reconst}} + \mathcal{L}_{\text{GAN}})$.
3. For learning θ_c, maximize $\mathcal{L}_{\text{GAN}}$.

In the following, we define $\mathcal{L}_{\text{reconst}}$ and $\mathcal{L}_{\text{GAN}}$, while the specification of $\mathcal{L}_{\text{sparsity}}$ is deferred to Sec. 6.

Reconstruction loss $\mathcal{L}_{\text{reconst}}$: The standard practice in learning encoder-decoder networks is to use the Euclidean distance between the input and decoded output, $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2$, for estimating the reconstruction error. However, recent findings demonstrate shortcomings of this practice [19]. Hence, instead, we define $\mathcal{L}_{\text{reconst}}$ based on the hidden representation in cLSTM – specifically, the output of the last hidden layer of cLSTM, $\phi(\boldsymbol{x})$, for input $\boldsymbol{x}$. Note that while $\boldsymbol{x}$ is a sequence of features, $\phi(\boldsymbol{x})$ represents a compact feature vector, capturing long-range dependencies in the input sequence. Therefore, it seems more appropriate to use $\phi(\boldsymbol{x})$, rather than $\boldsymbol{x}$, for specifying $\mathcal{L}_{\text{reconst}}$.

Specifically, we formulate $\mathcal{L}_{\text{reconst}}$ as an expectation of a log-likelihood $\log p(\phi(\boldsymbol{x})|\boldsymbol{e})$, given that $\boldsymbol{x}$ has been passed through the frame selector $\boldsymbol{s}$ and eLSTM, resulting in $\boldsymbol{e}$:

$$\mathcal{L}_{\text{reconst}} = \mathbb{E}[-\log p(\phi(\boldsymbol{x})|\boldsymbol{e})], \tag{3}$$

where expectation $\mathbb{E}$ is approximated as the empirical mean of training examples. In this paper, we consider $p(\phi(\boldsymbol{x})|\boldsymbol{e})) \propto \exp(-\|\phi(\boldsymbol{x}) - \phi(\hat{\boldsymbol{x}})\|^2)$, while other non-Gaussian likelihoods are also possible.

Loss of GAN, $\mathcal{L}_{\text{GAN}}$: Following [19], our goal is to train the discriminator such that cLSTM classifies reconstructed feature sequences $\hat{\boldsymbol{x}}$ as 'summary' and original feature sequences $\boldsymbol{x}$ as 'original'. In order to regularize this training, we additionally enforce that cLSTM learns to classify randomly generated summaries $\hat{\boldsymbol{x}}_p$ as 'summary', where $\hat{\boldsymbol{x}}_p$ is reconstructed from a subset of video frames randomly selected by sampling from a given prior distribution. In this paper, for this prior, we consider the uniform distribution. This gives:

$$\begin{aligned}\mathcal{L}_{\text{GAN}} = \; & \log(\text{cLSTM}(\boldsymbol{x})) + \log(1 - \text{cLSTM}(\hat{\boldsymbol{x}})) \\ & + \log(1 - \text{cLSTM}(\hat{\boldsymbol{x}}_p)),\end{aligned} \tag{4}$$

where $\text{cLSTM}(\cdot)$ denotes the binary soft-max output of cLSTM.

Given the above definitions of $\mathcal{L}_{\text{reconst}}$ and $\mathcal{L}_{\text{GAN}}$, as well as $\mathcal{L}_{\text{sparsity}}$ explained in Sec. 6, we update the parameters θ_s, θ_e, θ_d and θ_c using the Stochastic Gradient Variational Bayes estimation [17, 16], adapted for recurrent networks

[3]. Algorithm 1 summarizes all steps of our training. Note that Algorithm 1 uses capital letters to denote a mini-batch of the corresponding variables with small-letter notation in the previous text.

Algorithm 1 Training SUM/GAN model

Input: Training video sequences
Output: Learned parameters $\{\theta_s, \theta_e, \theta_d, \theta_c\}$.
Initialize all parameters $\{\theta_s, \theta_e, \theta_d, \theta_c\}$
for max number of iterations **do**
  $X \leftarrow$ mini-batch from CNN feature sequences
  $S \leftarrow \text{sLSTM}(X)$ % *select frames*
  $E = \text{eLSTM}(X, S)$ % *encoding*
  $\hat{X} = \text{dLSTM}(E)$ % *reconstruction*
  $S_p \leftarrow$ draw samples form the uniform distribution
  $E_p = \text{eLSTM}(X, S_p)$ % *encoding*
  $X_p = \text{dLSTM}(E_{S_p})$ % *reconstruction*
  % *Updates using Stochastic Gradient:*
  $\{\theta_s, \theta_e\} \overset{+}{\leftarrow} -\nabla(\mathcal{L}_{\text{reconst}} + \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{sparsity}})$
  $\{\theta_d\} \overset{+}{\leftarrow} -\nabla(\mathcal{L}_{\text{reconst}} + \mathcal{L}_{\text{GAN}})$
  $\{\theta_c\} \overset{+}{\leftarrow} +\nabla(\mathcal{L}_{\text{GAN}})$ % *maximization update*
end for

6. Variants of our Approach

This section explains our regularization of learning. We use the following three types of regularization, which define the corresponding variants of our approach.

Summary-Length Regularization penalizes having a large number of key frames selected in the summary as:

$$\mathcal{L}_{\text{sparsity}} = \left\| \frac{1}{M} \sum_{t=1}^{M} s_t - \sigma \right\|_2 \tag{5}$$

where M is the total number of video frames, and σ is an input hyper-parameter representing a percentage of frames that we expect to be selected in the summary. When our approach uses $\mathcal{L}_{\text{sparsity}}$, we call it SUM-GAN.

Diversity Regularization enforces selection of frames with high visual diversity, in order to mitigate redundancy in the summary. In this paper, we use two standard definitions for diversity regularization – namely, (i) Determinantal Point Process (DPP) [38, 7, 43]; and (ii) Repelling regularizer (REP) [45].

Following [43], our DPP based regularization is defined as:

$$\mathcal{L}^{\text{dpp}}_{\text{sparsity}} = -\log(P(\boldsymbol{s})) \tag{6}$$

where $P(\boldsymbol{s})$ is a probability that DPP assigns to the selection indicator $\boldsymbol{s}$. We compute $P(\boldsymbol{s}; L) = \frac{\det(L(\boldsymbol{s}))}{\det(L+I)}$, where L is an $M \times M$ similarity matrix between every two hidden states in eLSTM, I is an identity matrix and $L(\boldsymbol{s})$ is a smaller square matrix, cut down from L given $\boldsymbol{s}$. Let $\boldsymbol{e}_t$ be the hidden state of eLSTM at time t. For time steps t and t' the pairwise similarity values are defined as $L_{t,t'} = s_t s_{t'} \boldsymbol{e}_t \boldsymbol{e}_{t'}$.

When our approach uses $\mathcal{L}^{\text{dpp}}_{\text{sparsity}}$, we call it SUM-GAN$_{\text{dpp}}$.

For repelling regularization, we define

$$\mathcal{L}^{\text{rep}}_{\text{sparsity}} = \frac{1}{M(M-1)} \sum_{t} \sum_{t' \neq t} \left(\frac{\boldsymbol{e}_t^\top \boldsymbol{e}_{t'}}{\|\boldsymbol{e}_t\| \|\boldsymbol{e}_{t'}\|} \right) \tag{7}$$

and call this variant of our approach as SUM-GAN$_{\text{rep}}$.

Keyframe Regularization is specified for the supervised setting where ground-truth annotations of key frames are provided in training. This regularization enables a fair comparison of our approach with recently proposed supervised methods. Note that we here consider importance scores as 2D softmax outputs $\{\boldsymbol{s}_t\}$, rather than scalar values as introduced in Sec. 4. We define the sparsity loss as the cross-entropy loss:

$$\mathcal{L}^{\text{sup}}_{\text{sparsity}} = \frac{1}{M} \sum_{t} \text{cross-entropy}(\boldsymbol{s}_t, \hat{\boldsymbol{s}}_t). \tag{8}$$

We call this variant of our approach as SUM-GAN$_{\text{sup}}$.

7. Results

Datasets. We evaluate our approach on four datasets: SumMe [10], TVSum [34], Open Video Project (OVP) [24, 2], and Youtube [2]. 1) SumMe consists of 25 user videos. The videos capture multiple events such as cooking and sports. The video contents are diverse and include both first-person and third-person camera. The video lengths vary from 1.5 to 6.5 minutes. The dataset provides frame-level importance scores. 2) TVSum contains 50 videos from YouTube. The videos are selected from 10 categories in the TRECVid Multimedia Event Detection (MED) (5 videos per category). The video lengths vary from 1 to 5 minutes. Similar to SumMe, the video contents are diverse and include both ego-centric and third-person camera. 3) For OVP, we evaluate on the same 50 videos used in [2]. The videos are from various genres (e.g. documentary, educational) and their lengths vary from 1 to 4 minutes. 4) The YouTube dataset includes 50 videos collected from websites. The duration of the videos are from 1 to 10 minutes and the content include cartoons, news and sports.

Evaluation Setup. For a fair comparison with the state of the art, the keyshot-based metric proposed in [43] is used for evaluation. Let A be the generated keyshots and B the user-annotated keyshots. The precision and recall are defined based on the amount of temporal overlap between A and B as follows:

$$\begin{aligned} \text{precison} &= \frac{\text{duration of overlap between } A \text{ and } B}{\text{duration of } A} \\ \text{recall} &= \frac{\text{duration of overlap between } A \text{ and } B}{\text{duration of } B} \end{aligned} \tag{9}$$

Finally, the harmonic mean F-score is used as the evaluation metric. We follow the steps in [43] to convert frame-level scores to key frames and key shot summaries, and vice versa in all datasets. To generate key shots for datasets which only provide key frame scores, the videos are initially temporally segmented into disjoint intervals using KTS [26]. The resulting intervals are ranked based on their importance score where the importance score of an interval is equal to the average score of the frames in that interval. A subset of intervals are selected from the ranked intervals as keyshots such that the total duration of the generated key shots are less than 15% of the duration of the original video.

For datasets with multiple human annotations (in the form of key shots or key frames), we follow the standard approach described in [11, 34, 43] to create a single ground-truth set for evaluation. While evaluating our SUM-GAN$_{\text{sup}}$ model, we used the same train, test and validation split as in [43]. For fair comparison, we run it for five different random splits and report the average performance.

Implementation Details: For fair comparison with [43], we choose to use the output of pool5 layer of the GoogLeNet network [38] (1024-dimensions), trained on ImageNet [32], for the feature descriptor of each video frame. We use a two-layer LSTM with 1024 hidden units at each layer for discriminator LSTM (cLSTM). We use two two-layer LSTMs with 2048 hidden units at each layer for eLSTM and dLSTM respectively. It is shown in [35] that a decoder LSTM which attempts to reconstruct the reverse sequence is easier to train. Similarly, our dLSTM reconstruct the feature sequence in the reverse order. Note that while presenting $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ as the cLSTM input, both sequences should have similar ordering in time.

We initialize the parameters of eLSTM and dLSTM, with the parameters of a pre-trained recurrent autoencoder model trained on feature sequences from original videos. We find out that this helps to improve the overall accuracy and also results in faster convergence.

The sLSTM network is a two-layer bidirectional LSTM with 1024 hidden units. The output is a 2-dimensional softmax layer in the case of SUM-GAN$_{\text{sup}}$. We train our framework with Adam optimizer using the default parameters.

Baselines: It is important to point out that considering the generative structure of our approach and the definition of the update rules in Alg. 1, it is not possible to entirely replace subnetworks of our model baselines. Instead, in addition to different variations of our approach defined in sec. 6, we also evaluate the following baselines:

Method	SumMe	TVSum	OpenVideo	YouTube
SUM-GAN	38.7	50.8	71.5	58.9
SUM-GAN$_{\text{bas}}$	35.7	50.1	69.8	57.1
SUM-GAN$_{\text{w/o-GAN}}$	34.6	49.5	69.3	56.9
SUM-GAN$_{\text{w/o-}\boldsymbol{s}_p}$	37.2	50.4	71.5	58.4
SUM-GAN$_{\text{rep}}$	38.5	51.9	72.3	59.6
SUM-GAN$_{\text{dpp}}$	39.1	51.7	72.8	60.1
SUM-GAN$_{\text{sup}}$	41.7	56.3	77.3	62.5

Table 1: Comparison of different variations of our generative video summarization on benchmark datasets. The result for SUM-GAN is reported for $\sigma = 0.3$.

1) SUM-GAN$_{\text{bas}}$ which does not use the sparsity regularization,

2) SUM-GAN$_{\text{w/o-GAN}}$ which does not include $\mathcal{L}_{\text{GAN}}$ while updating $\{\theta_d\}$,

3) SUM-GAN$_{\text{w/o-}\boldsymbol{s}_p}$ which does not consider random summaries while training GAN, i.e. it replaces (4) with the followings:

$$\mathcal{L}_{\text{GAN}} = \quad \log(\text{cLSTM}(\boldsymbol{x})) + \log(1 - \text{cLSTM}(\hat{\boldsymbol{x}})).$$

7.1. Quantitative Results

Table 1 summarizes the accuracy of different variations of our approach. As is expected, the model with additional frame-level supervision, SUM-GAN$_{\text{sup}}$, outperforms the unsupervised variants by (2-5$\%$).

One interesting observation is that although explicit regularization of the model with 'diversity regularizers' (SUM-GAN$_{\text{dpp}}$ and SUM-GAN$_{\text{rep}}$) performs slightly better than the variant of our model with 'length regularizer' (SUM-GAN) the difference is not statistically significant. Furthermore, in the case of SumMe, SUM-GAN performs better than SUM-GAN$_{\text{rep}}$. This is particularly important because it verifies our main hypothesis that a good summary should include a subset of frames which provide similar content representation as of the original frame sequences. This suggests that if we constrain the summary to be shorter in length, implicitly the frames will be diverse. We also observe that SUM-GAN$_{\text{dpp}}$ performs better than SUM-GAN$_{\text{rep}}$ in all four datasets. We believe that this is mainly because of the fact that unlike the repelling regularizer, DPP is non-linear and can reinforce stronger regularization. Comparing the accuracy of SUM-GAN$_{\text{w/o-GAN}}$ with SUM-GAN shows that training with the combined losses from the VAE and GAN improves the accuracy.

We are particularly interested in comparing our performance in contrast with prior unsupervised and supervised methods. This comparison is presented in table 2. As shown, our unsupervised SUM-GAN$_{\text{dpp}}$ model outperforms all unsupervised approaches in all datasets. For SumMe, our approach is almost 5% better than the state-of-the-art unsupervised approaches. More importantly, the accuracy

Method	SumMe	TVSum	OpenVideo	YouTube
[2]	33.7	-	70.3	59.9
[21]	26.6	-	-	-
[15]	-	36.0	-	-
[34]	26.6	50.0	-	-
[4]	-	-	63.4	-
[23]	-	-	57.6	-
[44]	-	46.0	-	-
SUM-GAN$_{dpp}$	**39.1**	**51.7**	**72.8**	**60.1**

(a) Unsupervised Approaches

Method	SumMe	TVSum	OpenVideo	YouTube
[11]	39.7	-	-	-
[42]	40.9	-	76.6	60.2
[10]	39.3	-	-	-
[43]	38.6	54.7	-	-
[7]	-	-	**77.7**	60.8
SUM-GAN$_{sup}$	**41.7**	**56.3**	77.3	**62.5**

(b) Supervised Approaches

Table 2: Comparison of our proposed video summarization approach compared to state of the art. The reported results from the state of the art are from published results. Note that [42, 7] use only 39 sequences of non-cartoon videos.

Method	SumMe	TVSum
[42]	40.9	-
[43]	42.9	59.6
SUM-GAN	41.7	58.9
SUM-GAN$_{rep}$	42.5	59.3
SUM-GAN$_{dpp}$	43.4	59.5
SUM-GAN$_{sup}$	43.6	61.2

Table 3: Comparison of different variations of our generative video summarization with the state of the art for SumMe and TVSum datasets when the training data is augmented with videos from OVP and YouTube datasets. For [43], results w/o domain adaptation are reported

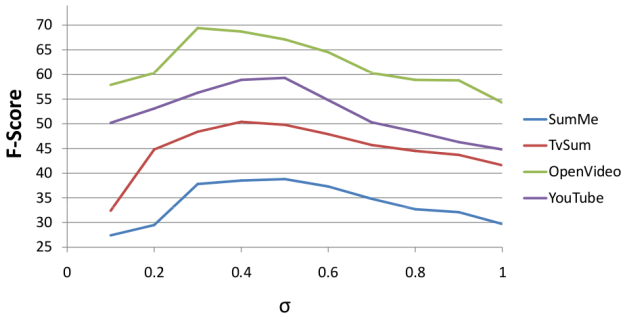


Figure 4: F-score results for different values of σ on SumMe, TvSum, OpenVideo, and YouTube.

Method	SumMe	TVSum
[34]	-	50.0
[42]	-	60.0
[43]	38.1	54.0
SUM-GAN	37.8	53.2
SUM-GAN$_{rep}$	38.8	54.1
SUM-GAN$_{dpp}$	41.2	53.9
SUM-GAN$_{sup}$	39.5	59.5

Table 4: Comparison of different variations of our generative video summarization with the state of the art for SumMe and TVSum datasets when using shallow features.

of SUM-GAN$_{dpp}$ is comparably close to the supervised methods in TVSum, OVP and YouTube datasets.

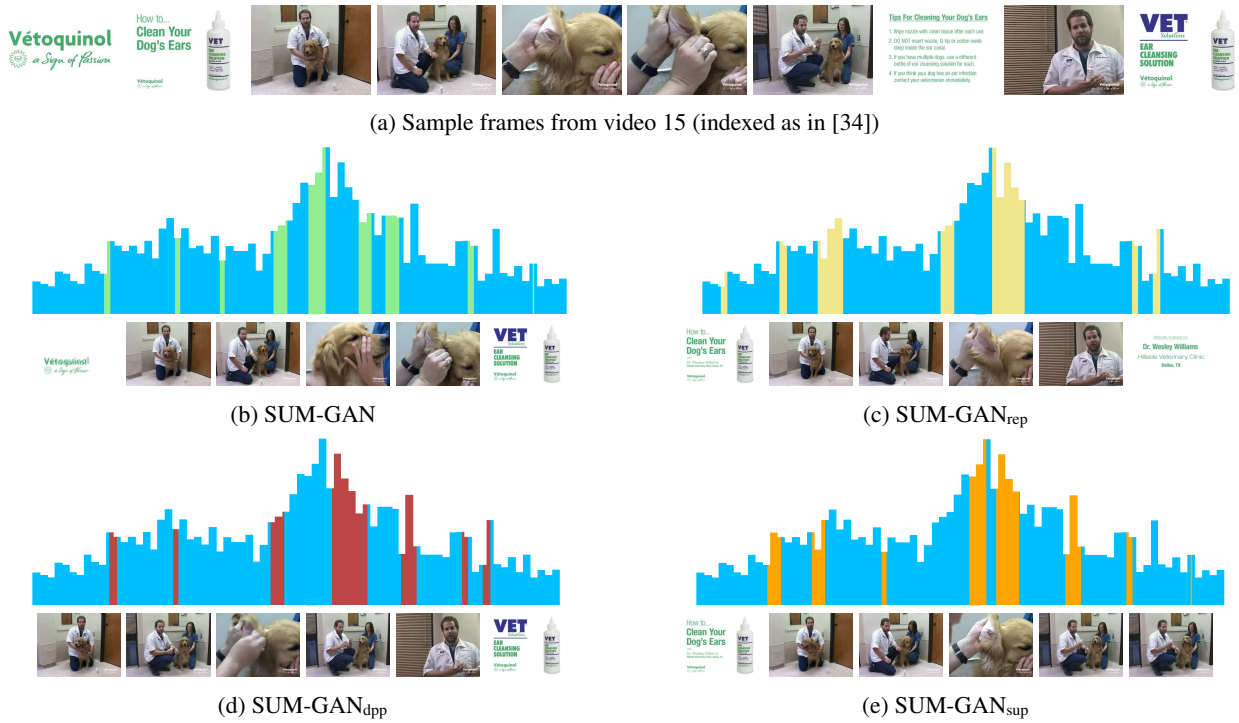
Comparing with the state-of-the-art supervised approaches, our supervised variant, SUM-GAN$_{sup}$, outperforms in all datasets except OVP. Even in the case of OVP, we are statistically close to the best reported accuracy with 0.4% margin. We hypothesize that the accuracy boost is mainly because of the additional learning signal from the cLSTM. Note that the discriminator observes a longer sequence and classifies based on a learned semantic representation of the feature sequence. This enables the discriminator to provide a more informative signal regarding the importance of the frames for content similarity.

Zhang et al. [43] augment the SumMe and TVSum datasets with OVP and YouTube datasets and improve the accuracy on SumMe and TVSum. Table 3 shows the accuracy results in comparison with results reported in [43] when training dataset is augmented. Except for SUM-GAN$_{sup}$, which we use 80% of the target dataset in training, for the unsupervised variants of our approach we use all four datasets in training. The most important observation is that one of our unsupervised variations, SUM-GAN$_{dpp}$, outperforms the state of the art in SumMe. This shows that if trained with more unsupervised video data, our model is able to learn summaries which are competitive with the models trained using key frame annotations.

Finally, we evaluate the performance of our approach for different percentages of σ values for our SUM-GAN model. Fig. 4 shows the resulting F-score values for different σ's on four different datasets. While the performance is consistent for $0.3 \leq \sigma \leq 0.5$, it drops rapidly as $\sigma \rightarrow 1$ or $\sigma \rightarrow 0$.

7.2. Comparison with Shallow Features

We verified the generalizability of our video summarization approach to non-deep features by evaluating our model



(a) Sample frames from video 15 (indexed as in [34])

(b) SUM-GAN

(c) SUM-GAN$_{\text{rep}}$

(d) SUM-GAN$_{\text{dpp}}$

(e) SUM-GAN$_{\text{sup}}$

Figure 5: Example summaries from a sample video in TvSum [34]. The blue bars show the annotation importance scores. The colored segments are the selected subset of frames using the specified method.

with the shallow features employed in [42, 43]. Table 4 shows the performance of our model compared to the state-of-the-art models which use shallow features. Besides the reported results in [42] for TvSum, where the shallow features outperform the deep features, our model consistently performs better the state of the art. Unlike [42], our model grounded on deep features still performs better than the same model grounded on shallow features.

7.3. Qualitative Results

To better illustrate the temporal selection pattern of different variations of our approach, we demonstrate the selected frames on an example video in Fig. 5. The blue background shows the frame-level importance scores. The colored regions are the selected subsets for different methods. The visualized key frames for different variants supports the result presented in Table 1. Despite small variations, all four approaches cover the temporal regions with high frame-level score. Most of the failure cases occurred in videos which consist of frames with very slow motions and no scene-change.

8. Conclusion

We propose a generative architecture based on variational recurrent auto-encoders and generative adversarial networks for unsupervised video summarization to select a subset of key frames. The main hypothesis is that the learned representation of the summary video and the original video should be similar. The summarizer aims to summarize the video such that the discriminator is fooled and the discriminator aims to recognize the summary videos from original videos. The entire model is trained in an adversarial manner where the GAN's discriminator is used to learn a discrete similarity measure for training the recurrent encoder/decoder and the frame selector LSTMs. Variations of our approach are defined using different regularizations. Evaluation on benchmark datasets show that all unsupervised variations of our approach outperform the state of the art in video summarization by 2-5% and provides a comparable accuracy to the state-of-the-art supervised approaches. We also verified that the supervised variation of our approach outperforms the state of the art by 1-4%.

Acknowledgement

This work was supported in part by DARPA XAI and NSF RI1302700.

References

[1] A. Aner and J. R. Kender. *Video Summaries through Mosaic-Based Shot and Scene Clustering*, pages 388–402. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[2] S. E. F. De Avila, A. P. B. Lopes, A. da Luz, and A. de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

[3] O. Fabius and J. R. van Amersfoort. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014.

[4] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini. Stimo: Still and moving video storyboard for the web scenario. *Multimedia Tools and Applications*, 46(1):47–69, 2010.

[5] A. Ghosh, V. Kulharia, A. Mukerjee, V. Namboodiri, and M. Bansal. Contextual rnn-gans for abstract reasoning diagram generation. *arXiv preprint arXiv:1609.09444*, 2016.

[6] D. B. Goldman, B. Curless, S. M. Seitz, and D. Salesin. Schematic storyboarding for video visualization and editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3), July 2006.

[7] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *NIPS*, pages 2069–2077, 2014.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[9] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *ECCV*, 2014.

[10] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *ECCV*, pages 505–520. Springer, 2014.

[11] M. Gygli, H. Grabner, and L. Van Gool. Video summarization by learning submodular mixtures of objectives. In *CVPR*, pages 3090–3098, 2015.

[12] T. Halperin, Y. Poleg, C. Arora, and S. Peleg. Egosampling: Wide view hyperlapse from single and multiple egocentric videos. *CoRR*, abs/1604.07741, 2016.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] N. Joshi, W. Kienzle, M. Toelle, M. Uyttendaele, and M. F. Cohen. Real-time hyperlapse creation via optimal frame selection. *ACM Trans. Graph.*, 34(4):63:1–63:9, July 2015.

[15] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *CVPR*, pages 2698–2705, 2013.

[16] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *ICLR*, 2014.

[17] D. P. Kingma and M. Welling. Stochastic gradient vb and the variational auto-encoder. *Talk Slides*, 2014.

[18] J. Kopf, M. F. Cohen, and R. Szeliski. First-person hyperlapse videos. *ACM Trans. Graph.*, 33(4):78:1–78:10, July 2014.

[19] A. B. L. Larsen, S. K. Sønderby, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[20] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, pages 1346–1353, 2012.

[21] Y. Li and B. Merialdo. Multi-video summarization based on video-mmr. In *WIAMIS*, pages 1–4. IEEE, 2010.

[22] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[23] P. Mundur, Y. Rao, and Y. Yesha. Keyframe-based video summarization using delaunay clustering. *International Journal on Digital Libraries*, 6(2):219–232, 2006.

[24] Open video project. http://www.open-video.org.

[25] Y. Poleg, T. Halperin, C. Arora, and S. Peleg. Egosampling: Fast-forward and stereo for egocentric videos. In *CVPR*, June 2015.

[26] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, Zurich, Switzerland, Sept. 2014. Springer.

[27] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg. Webcam synopsis: Peeking around the world. In *ICCV*, pages 1–8, Oct 2007.

[28] Y. Pritch, A. Rav-Acha, and S. Peleg. Nonchronological video synopsis and indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1971–1984, Nov 2008.

[29] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[30] A. Rav-Acha, Y. Pritch, and S. Peleg. Making a long video short: Dynamic video synopsis. In *CVPR*, pages 435–441, 2006.

[31] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, pages 1–42, April 2015.

[33] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*, 2016.

[34] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, pages 5179–5187, 2015.

[35] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *CoRR, abs/1502.04681*, 2, 2015.

[36] Y. Statistics. `https://www.youtube.com/yt/press/statistics.html`. Accessed: 2016.

[37] M. Sun, A. Farhadi, B. Taskar, and S. Seitz. *Salient Montages from Unconstrained Videos*, pages 472–488. Springer International Publishing, Cham, 2014.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[39] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. *arXiv preprint arXiv:1609.02612*, 2016.

[40] H. wen Kang, Y. Matsushita, X. Tang, and X. quan Chen. Space-time video montage. In *CVPR*, pages 1331–1338, 2006.

[41] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *ICCV*, pages 4633–4641, 2015.

[42] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In *CVPR*, June 2016.

[43] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In *ECCV*, pages 766–782. Springer, 2016.

[44] B. Zhao and E. P. Xing. Quasi real-time summarization for consumer videos. In *CVPR*, pages 2513–2520, 2014.

[45] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.