AC-SUM-GAN: Connecting Actor-Critic and Generative Adversarial Networks for Unsupervised Video Summarization

Evlampios Apostolidis, Eleni Adamantidou, Alexandros I. Metsai, Vasileios Mezaris, *Senior Member, IEEE*, and Ioannis Patras, *Senior Member, IEEE*

***Abstract*— This paper presents a new method for unsupervised video summarization. The proposed architecture embeds an Actor-Critic model into a Generative Adversarial Network and formulates the selection of important video fragments (that will be used to form the summary) as a sequence generation task. The Actor and the Critic take part in a game that incrementally leads to the selection of the video key-fragments, and their choices at each step of the game result in a set of rewards from the Discriminator. The designed training workflow allows the Actor and Critic to discover a space of actions and automatically learn a policy for key-fragment selection. Moreover, the introduced criterion for choosing the best model after the training ends, enables the automatic selection of proper values for parameters of the training process that are not learned from the data (such as the regularization factor σ). Experimental evaluation on two benchmark datasets (SumMe and TVSum) demonstrates that the proposed AC-SUM-GAN model performs consistently well and gives SoA results in comparison to unsupervised methods, that are also competitive with respect to supervised methods.**

***Index Terms*— Video summarization, unsupervised machine learning, actor-critic model, reinforcement learning, generative adversarial networks.**

I. Introduction

NOWADAYS, we are witnessing a tremendous growth of online-available video material, that is fueled mainly by two factors: i) the constantly increasing engagement of users with smart devices that carry powerful video recording sensors and online content sharing functionalities, and ii) the wide-spread use of video sharing platforms (e.g., YouTube, Vimeo, DailyMotion) and social networks (e.g., Facebook, Twitter, Instagram) as communication means of both amateur and professional users (such as media organizations, news agencies and advertising companies). This growth has rapidly increased the need for technologies that facilitate users' navigation within vast and constantly-increasing collections of videos, and the quick retrieval of the piece of video content that they are looking for. Part of the response to this demand is the development of techniques for automatic video summarization. These methods generate a concise synopsis that conveys the important parts of the full-length video; based on this, viewers can have a quick overview of the whole story without having to watch the entire content.

Several approaches were proposed over the last couple of decades to automate video summarization, and the current SoA is represented by deep-learning-based methods. A coarse division of these methods can be made between supervised and unsupervised approaches, and a more detailed classification is shown in Fig. 1; this taxonomy will be the basis for presenting the relevant literature in Section II. In this figure, we also show the positioning of the proposed AC-SUM-GAN method, in relation to past works.

Supervised methods rely on datasets with ground-truth human-generated summaries (e.g., SumMe [1] and TVSum [2]), based on which they try to discover the underlying criterion for video summarization. However, the generation of ground-truth data (usually in the form of video summaries or annotations indicating the importance of video frames) is a time-consuming and tedious task. Moreover, the subjectivity of video summarization can lead to quite different summaries for the same video, thus making it hard to train a method using these summaries as ground-truth.

Unsupervised approaches try to learn video summarization without the use of ground-truth data. Some of them rely on Generative Adversarial Networks (GANs) to find a way to assess the representativeness of any created summary. Others build on reinforcement learning and define rewards based on the desired characteristics of the video summary, such as the

Manuscript received July 23, 2020; revised September 30, 2020; accepted November 1, 2020. Date of publication November 16, 2020; date of current version August 4, 2021. This work was supported by the EU's Horizon 2020 Research and Innovation Programme under Grant H2020-780656 ReTV. The work of Ioannis Patras was supported by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/R026424/1. This article was recommended by Associate Editor Z.-J. Zha. *(Corresponding author: Evlampios Apostolidis.)*

Evlampios Apostolidis is with the Centre for Research and Technology Hellas, Information Technologies Institute, 570 01 Thessaloniki, Greece, and also with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: apostolid@iti.gr).

Eleni Adamantidou, Alexandros I. Metsai, and Vasileios Mezaris are with the Centre for Research and Technology Hellas, Information Technologies Institute, 570 01 Thessaloniki, Greece (e-mail: adamelen@iti.gr; alexmetsai@iti.gr; bmezaris@iti.gr).

Ioannis Patras is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: i.patras@qmul.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2020.3037883.

Digital Object Identifier 10.1109/TCSVT.2020.3037883

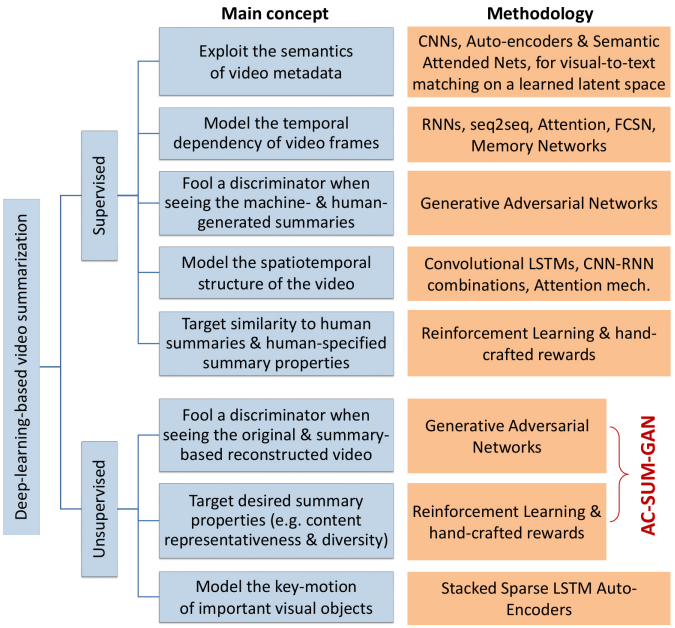


Fig. 1. A taxonomy of the current SoA methods for video summarization, and the positioning of the proposed AC-SUM-GAN method.

diversity of its visual content. Most of them utilize Long Short-Term Memory (LSTM) units [3] to learn how to assess the importance of each video frame. However, experimentation with some of these methods (dppLSTM [4], DR-DSN [5], SUM-GAN-sl [6], SUM-GAN-AAE [7]) resulted in findings that are consistent with the claims in [8] about the low variation of the computed frame-level importance scores by LSTMs. As a consequence, the selections made by the trained LSTM seem to have a limited impact in summarization; the latter is mainly affected by factors such as the video fragmentation, or the approach used for fragment selection given a target summary length (such as the Knapsack algorithm).

To address the above limitations, we formulate the selection of important video fragments - that will be subsequently used to define the video key-fragments and create a summary of a given length using the Knapsack algorithm - as a sequence generation task and propose a method for video summarization, where an Actor-Critic (AC) model is embedded into a GAN. Different from other GAN-based approaches for unsupervised video summarization (e.g., [6]–[10]) that use the Discriminator's feedback to optimize the key-frame/fragment selector, in our method the Discriminator's feedback is used to train the Actor-Critic model, which learns a value function (Critic) and a policy for key-fragment selection (Actor). The proposed approach is fully unsupervised; thus, it overcomes the need for expensive and laborious human annotations, and the use of ground-truth data. Moreover, it eliminates the need for external supervision or hand-crafted rewards, as it automatically learns a policy for key-fragment selection, based on the feedback of a trainable Discriminator. Finally, we introduce a criterion for model selection after the end of training, which allows the proper configuration of parameters of the training process in a fully unsupervised and automatic manner. We should note that combining AC and GAN was discussed only very recently for other tasks [11], and our work is the first to propose this for video summarization. We show experimentally that the use of the AC model, as proposed, leads to competitive performance even compared to SoA supervised video summarization methods.

TABLE I
LIST OF ACRONYMS

Acronym	Explanation
AC model	Actor-Critic model
CNN	Convolutional Neural Network
CSNet	Chunk and Stride Network
DCNN	Deep Convolutional Neural Network
DTR unit	Dilated Temporal Relational unit
FCSN	Fully-Convolutional Sequence Network
GAN	Generative Adversarial Network
KTS method	Kernel Temporal Segmentation method
LSTF	Long-Short-Term Features
LSTM unit	Long Short-Term Memory unit
MLP	Multi-Layer Perceptron
RNN	Recurrent Neural Network
VAE	Variational Auto-Encoder

Our contributions can be summarized as follows:

- We introduce the use of the AC model for reinforcement learning to address the task of video summarization;
- We propose a novel architecture that embeds the AC model into a GAN to learn a policy for key-fragment selection and summarization in a fully unsupervised manner;
- We examine the use of different criteria for unsupervised model selection based on the training set and after the end of the model's training process.

To facilitate reading, in Table I we provide a list of acronyms used in the sequel and their explanations.

II. Related Work

A. Supervised Video Summarization

Early supervised video summarization approaches build on the advances of CNN/DCNN architectures to extract the semantics of the visual content and perform semantic-driven summarization. To this direction, a couple of methods perform summarization by learning importance [12] or transferring the summary structure [13] from semantically-similar videos. Panda *et al.* [14] use video metadata for video categorization and to learn what is important in each category, and perform category-driven summarization by maximizing the relevance between the summary and the video's category. Similarly, a few methods [15]–[17] learn category-driven summarization in various ways, e.g., by using action classifiers. Otani *et al.* [18] and Yuan *et al.* [19] define a summary by maximizing its relevance with the video metadata, after projecting visual and textual data in a common latent space. Finally, Wei *et al.* [20] apply a visual-to-text mapping and a semantic-based key-fragment selection using semantic attended networks. However, most of the above methods examine only the visual cues and do not consider the sequential structure of the video. Hence, they might erroneously ignore video parts that are useful for providing a complete summary of the story, due to their resemblance with parts already included in the summary.

To tackle the aforementioned shortcoming, a few methods cast video summarization as a structured prediction problem and model the temporal structure of the video and the temporal dependency among video frames to estimate their importance. The first approach to this direction [4], uses an LSTM to model variable-range dependency among frames, and estimates their importance using a multi-layer perceptron (MLP). Zhao *et al.* [21] propose a two-layer LSTM architecture to extract and encode data about the video structure (first layer), and define the key-fragments of the video (second layer). The previous method is extended in [22] to identify and exploit the shot-level temporal structure of the video. Casas and Koblents [23] extend [4] by introducing an attention mechanism to model the evolution of the users' interest. In the same direction, a few methods utilize sequence-to-sequence (a.k.a. seq2seq) architectures in combination with attention mechanisms. Fajtl *et al.* [24] present a seq2seq network made of a soft self-attention mechanism and a two-layer fully connected network for regression of the frames' importance scores. Ji *et al.* [25] propose an LSTM-based Encoder-Decoder network with an intermediate attention layer. Liu *et al.* [26] employ a Generator-Discriminator architecture (similar to the one in [9]) as an internal mechanism to estimate the representativeness of each shot and define a set of candidate key-frames, and then they use a multi-head attention model to select the key-frames that form the summary. Rochan *et al.* [27] tackle video summarization as a semantic segmentation task and propose using a Fully-Convolutional Sequence Network (FCSN). Finally, to tackle issues related to the limited capacity of LSTMs, some techniques use additional memory ([28], [29]). For example, [29] stacks multiple LSTM and memory layers hierarchically to derive long-term temporal context.

Following a different approach to minimizing the distance between the machine-generated and the ground-truth summaries, a couple of methods use GANs. Zhang *et al.* [30] estimate the frames' dependency at different temporal windows using LSTMs and Dilated Temporal Relational units, and learn summarization by trying to fool a trainable Discriminator when distinguishing the machine summary from the ground-truth and a randomly-created one. Fu *et al.* [31] suggest an adversarial learning approach for semi-supervised video summarization; the Generator (an attention-based Pointer Network [32]) defines the boundaries of each video fragment that is used to form the summary; the Discriminator (a 3D-CNN classifier) judges whether a fragment is from a ground-truth or a machine summary. Instead of using the typical adversarial loss, the Discriminator's output is used as a reward to train the Generator via reinforcement learning.

Aiming to better learn how to estimate the importance of video frames/fragments, some techniques pay attention to both the spatial and temporal structure of the video. Lal *et al.* [33] present an Encoder-Decoder architecture with convolutional LSTMs that models the spatiotemporal relationship among parts of the video. Yuan *et al.* [34] use 3D-CNNs and convolutional LSTMs to model the spatiotemporal structure of the video and select the video key-frames, while Chu and Liu [35] extract spatial and temporal information by processing the raw frames and their optical flow maps with CNNs. Elfeki and Borji [36] combine CNNs and RNNs to form spatiotemporal feature vectors, that are then used to estimate the level of activity and importance of each frame. Huang and Wang [37] train a neural network for spatiotemporal data extraction and create an inter-frames motion curve; the latter is used by a self-attention mechanism that selects the key-frames/fragments of the video. Finally, the temporal dynamics and the spatial information of the visual content are jointly considered and modeled by long-short-term features (LSTF) in [38], to address the task of scene classification in videos; such features can be used to determine the key-frames/fragments of the video.

Contrary to the above approaches, the weakly-supervised video summarization algorithm of [39] uses the principles of reinforcement learning to learn summarization based on sparse human annotations and hand-crafted rewards. The former indicate the importance of a small subset of frames, while the latter relate to the similarity between the machine- and the human-selected fragments, as well as to specific characteristics of the created summary (e.g., its representativeness).

B. Unsupervised Video Summarization

To avoid using ground-truth-annotated training data for learning video summarization, most existing unsupervised approaches focus on the principle that a representative summary ought to assist the viewer to infer the original video content. Instead of defining hand-crafted thresholds with regards to the desired similarity between the generated summary and the original video, these techniques rely on GANs to reconstruct the original video using the defined summary, and thus to automatically find the minimum distance between the summary and the video in a learned latent space. Mahasseni *et al.* [9] are the first to combine an LSTM-based key-frame selector with a Variational Auto-Encoder (VAE) and a trainable Discriminator, and learn video summarization through an adversarial learning process that aims to minimize the distance between the original video and the summary-based reconstructed version of it. Apostolidis *et al.* [6] build on the network architecture of [9], and suggest a stepwise, label-based approach for training the adversarial part of the network, that leads to improved performance. Yung *et al.* [8] also rely on a VAE-GAN architecture but extend it with a chunk and stride network (CSNet) and a tailored attention mechanism for assessing temporal dependencies at different granularities for selecting the video key-frames. Yuan *et al.* [10] aim to maximize the mutual information between the summary and the video using a trainable couple of Discriminators and a cycle-consistent adversarial learning objective. Apostolidis *et al.* [7] introduce a variation of [6] that replaces the VAE with an Attention Auto-Encoder for learning an attention-driven reconstruction of the original video that subsequently improves the key-fragment selection process. Similarly, [40] presents a self-attention-based conditional GAN to simultaneously minimize the distance between the generated and raw frame features, and focus on the most important fragments of the video. Finally, [41] learns video summarization from unpaired data based on an adversarial process and an FCSN, and defines a mapping function of a raw video to a human-like summary.

Aiming to deal with the unstable training [5] and the restricted evaluation criteria of GAN-based methods (that mainly focus on the summary's ability to allow the reconstruction of the original video), some unsupervised approaches perform summarization by paying attention to specific properties of the video summary. To this direction, they utilize the principles of reinforcement learning in combination with hand-crafted reward functions that quantify the existence of desired characteristics in the generated summary. In this context, [5] formulates video summarization as a sequential decision-making process and trains a summarizer to produce diverse and representative video summaries using a diversity-representativeness reward. Gonuguntla *et al.* [42] utilize Temporal Segment Networks (proposed in [43] for action recognition in videos) to extract spatial and temporal information about the video frames, and train the summarizer through a reward function that assesses the preservation of the video's main spatio-temporal patterns in the produced summary. Zhao *et al.* [44] present a mechanism for video reconstruction and summarization. The former aims to estimate the extent to which the summary allows the viewer to infer the original video. The latter is learned based on the reconstructor's feedback and the output of models assessing the representativeness and diversity of the generated summary.

Building on a different basis, [45] focuses on the preservation in the summary of the underlying fine-grained semantic and motion information of the video. For this, it represents the whole video by creating super-segmented object motion clips, extracts the key motions of appearing objects, and uses an online motion auto-encoder model (Stacked Sparse LSTM Auto-Encoder) to memorize past states of object motions by continuously updating a tailored recurrent auto-encoder network. The trained model is finally used to generate summaries that present the representative objects in the video and the attractive actions made by each of these objects.

C. Relation of the Proposed Method With the Bibliography

Based on the above review of the current SoA on video summarization, we identify a number of connections between the introduced summarization algorithm and earlier works on this area. Similarly to [31], our method establishes a link between GANs and reinforcement learning approaches and uses the Discriminator's feedback to train the summarizer. However, our model is trained in a fully unsupervised manner and, thus, eliminates the need for human annotations. Given this observation, our technique is mostly associated with unsupervised algorithms for video summarization that rely on adversarial or reinforcement learning (see its positioning in Fig. 1). More specifically, the proposed model is an extension of the architecture from [9], which aims to overcome a limitation of LSTM-based algorithms for unsupervised video summarization. This limitation (discussed also in [8]) relates to the estimation of frame-level importance scores that exhibit very low variation, and thus have a restricted impact when selecting the video fragments that will form the summary (using e.g., the Knapsack algorithm). In contrast to these techniques, the developed algorithm selects the important parts of the video by introducing a trainable pair of models (Actor and Critic). The latter is capable of exploring a space of actions and of automatically learning a strategy that clearly indicates the important fragments of the video by boosting their importance score. In this way, the selected fragments have a key role when defining the video key-fragments and forming the summary, using the Knapsack algorithm. Moreover, contrary to existing summarization approaches relying on reinforcement learning, our method eliminates the need for hand-crafted rewards as it automatically learns a value function (Critic) that drives the optimal policy (Actor) for key-fragment selection, based on the Discriminator's feedback. Finally, the most important differences compared to our previous method [7] are: i) the use of an AC model for fragment selection instead of using an LSTM (for reasons discussed above) and ii) the use of a stochastic Variational Auto-Encoder for video reconstruction instead of using a deterministic Attention Auto-Encoder.

Besides the above discussed relation with literature works on video summarization, in terms of conceptualizing a link between AC and GANs our method is related to the works of [11], [46], [47]. Pfau and Vinyals [46] are the first to explore a connection between Actor-Critic and adversarial learning by interpreting GANs as Actor-Critic methods in an environment where the Actor cannot affect the reward. Goyal *et al.* [11] investigate this connection more thoroughly and empirically in the setting of natural language generation. Savioli [47] presents an approach that combines GANs with AC to train an Encoder-Decoder architecture for image compression of high-resolution images. Different to these works, we utilize the idea of an AC-GAN architecture to address the task of video summarization, and we embed an AC model into a GAN to learn a policy for key-fragment selection and summarization in a fully unsupervised manner.

Finally, with respect to previously published works in IEEE TCSVT, our manuscript is most closely related to [17], [19], [25], [37] that suggest different deep-learning-based approaches for supervised video summarization. However, differently from them, our manuscript proposes a method that: i) learns summarization in a fully unsupervised manner, and ii) is the first to introduce the integration of a trainable AC model into a GAN to learn a policy for key-fragment selection and summarization.

III. Proposed Approach

A. Formulation of the Video Summarization Task

The building blocks for defining a new formulation of the video summarization task were the works of [11] and [9]. The former discussed a connection between GANs and Actor-Critic models, as the core part of an algorithm that deals with language modeling tasks. The latter, was the first to utilize the generative adversarial learning for unsupervised video summarization, by introducing a trainable Discriminator to automatically define a similarity threshold between the original video and a reconstructed version of it based on a sparse set of selected key-frames (i.e., the video summary).

We transfer the idea of [11] to the visual domain and formulate the selection of important parts of the video (that

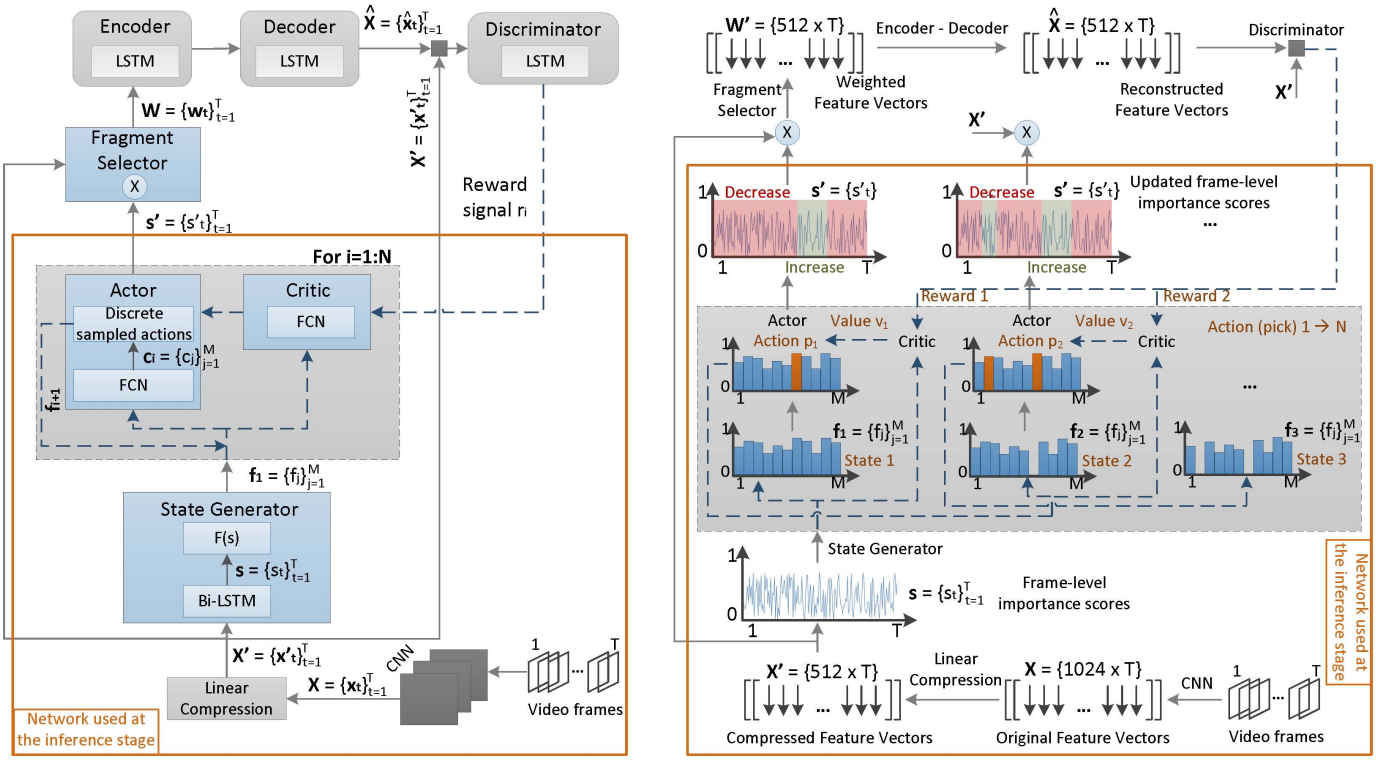


Fig. 2. The AC-SUM-GAN architecture. On the left side we show the building blocks of the architecture and their connections. Blue coloured rectangles indicate parts related to the Actor-Critic model. On the right side we give an example of the data flow by presenting the input and output of each different part of the architecture. On both sides of the figure, dashed lines represent iterative processes during the training of the AC part. The orange box shows the part of the architecture that is used for inference; at the training stage, the entire architecture is used.

will be used to define the video key-fragments and produce the summary using the Knapsack algorithm) as a "visual sentence" generation process. In most existing approaches for real-valued data sequence generation (e.g., text, speech or music synthesis [48]) the used vocabulary of tokens for synthesizing the data sequence is a predefined collection of e.g., letters, words, or music notes. In our conceptualized "visual sentence" generation process this vocabulary is created on-the-fly according to the visual content of the submitted video for summarization. In particular, the tokens of the created vocabulary when summarizing a video, correspond to video fragments of roughly the same length, where each fragment presents a different part of the story. Based on the above, we formulate video summarization as a sequential process that aims to progressively select a set of visual tokens and produce a "visual sentence" that conveys the essential parts and the flow of the story.

To materialize this formulation, we start from the unsupervised summarization algorithm of [9] and propose a new architecture, called AC-SUM-GAN, that embeds an Actor-Critic model into a Generative Adversarial Network to learn the optimal policy for selecting the video key-fragments and form the summary. The Actor has the role of the sequence generator and the generation is performed incrementally based on a set of discrete sampled actions over a group of video fragments. These actions indicate the selection or not of a fragment and affect the state of the action-state space that is essential for training the AC model, while the number of actions N is a hyper-parameter of the architecture, which relates to the duration of the generated summary. The Critic has the role of the evaluator of the Actor's choices and returns a value for scoring each choice according to its impact on the action-state space. Finally, the Discriminator acts as the AC environment and returns a reward that is used to train the Actor-Critic model, which learns a value function (Critic) and a policy for key-fragment selection (Actor). This reward relates to the appropriateness of the Actor's choices that define the video summary, for eventually reconstructing a video that is indistinguishable from the original one. In the sequel we describe in more detail the overall network architecture and the learning objectives and pipeline. With respect to the used notation: capital bold letters denote matrices, small bold letters denote vectors and non-bold letters (either capital or small) denote scalar values.

B. Overall Network Architecture

Figure 2 shows the architecture of the proposed AC-SUM-GAN model. The sub-figure on the left side provides details about the building blocks of the architecture and shows how these blocks are connected and interact. Blue coloured rectangles indicate parts related to the Actor-Critic model. The sub-figure on the right side presents the data flow in the architecture. These illustrations show the input and output of each different part of the architecture, thus explaining the role of each part of the architecture and the way that the AC model

is used to incrementally select the key-fragments of the video and form the summary. On both sides of Fig. 2, dashed lines represent iterative processes during the training of the AC part.

The proposed AC-SUM-GAN architecture extends [9] by: i) introducing an AC model for key-fragment selection, ii) adding a new component (called State Generator) that integrates the Frame Selector of [9] (bi-directional LSTM) and produces a state of a fixed length which is essential for training the AC model, and iii) using the Discriminator's feedback to automatically learn a value function (Critic) and a policy for key-fragment selection (Actor).

All the different components of the proposed architecture (see the left side of Fig. 2) are trained through the incremental 4-step process explained in Sec. III-C. After the end of the training, the model's components surrounded by the orange box in Fig. 2 are used for summarizing a new (i.e., unseen during training) video. At inference time, given a video of T frames, the model gets as input the CNN-based deep feature representations of the video frames ($\boldsymbol{X} = \{\boldsymbol{x_t}\}_{t=1}^{T}$) and produces a sequence of frame-level scores ($\boldsymbol{s'} = \{s'_t\}_{t=1}^{T}$) that signify each frame's importance and thus, its suitability to be included in the summary. This process starts by passing the deep feature vectors through a linear compression layer (fully connected layer for dimensionality reduction) that reduces their size. Then, the State Generator gets the compressed feature vectors and produces the initial state of the action-state space for training the AC model. For this, it assigns an importance score to every video frame according to its temporal dependency with the other frames of the video, and computes fragment-level importance scores via an average pooling operation. Given this state, the trained Actor plays an "N-picks" game and selects N non-overlapping, roughly equal in length, fragments of the video. The Actor's choices result to an update of the initially computed weights, by increasing the scores of the frame sequences corresponding to the selected fragments and reducing the scores of the remaining ones, according to predefined scaling factors. The updated sequence of frame-level scores - with the selected fragments being clearly indicated by greater scores - forms the output $\boldsymbol{s'}$ of the network's part that is used at the inference stage. This output $\boldsymbol{s'}$ is finally used to define a video summary that does not exceed the target summary duration (in most SoA summarization works this is typically set to 15% of the original video duration, a condition adopted also here to allow direct comparisons). For this, importance scores are computed at the level of video fragments defined using the KTS method [49], and the key-fragments of the video are selected and form the summary using the Knapsack algorithm.

In the sequel we present the different parts of the architecture by describing the training workflow. In particular, given a video of T frames and a linear compression layer that reduces the size of the deep feature vectors, the processing pipeline for training AC-SUM-GAN comprises of:

A **State Generator** that consists of a bi-directional LSTM followed by an average pooling operator. The former captures the temporal dependency over the sequence of frames in both forward and backward direction, and assigns a weight to each video frame that represents its importance (frame-level scores $\boldsymbol{s} = \{s_t\}_{t=1}^{T}$ with $s_t \in \mathbb{R}$ and $0 \leq s_t \leq 1$). The latter takes the computed frame-level scores $\boldsymbol{s}$ and produces the initial state $\boldsymbol{f}$ of the AC action-state space by calculating scores at a coarser fragment-level; for this, the video is segmented into M non-overlapping fragments of duration d, and a score is computed for each fragment by averaging the weights of the frames included in the fragment ($\boldsymbol{f} = \{f_j\}_{j=1}^{M}$ with $f_j \in \mathbb{R}$ and $f_j = (\sum_{t=(j-1)d+1}^{j\,d} s_t)/d$).

An **Actor** (fully connected network), who plays an "N-picks" game to explore the action-state space, and in every step i (with $1 \leq i \leq N$) of this game: i) gets the current state ($\boldsymbol{f_i} = \{f_j\}_{j=1}^{M}$), ii) produces a distribution of actions $\boldsymbol{c_i} = \{c_j\}_{j=1}^{M}$, and iii) takes an action p_i by sampling the computed distribution, and picks a video fragment k. This action leads to the next state $\boldsymbol{f_{i+1}}$ of the action-state space, which is produced by zeroing its k^{th} element ($f_k = 0$) to minimize the probability of having the k^{th} fragment re-selected in a subsequent step of the game. Moreover, it affects the computed frame-level weights $\boldsymbol{s}$ by increasing the ones associated to the frames within the selected fragment using action-weighting factors and reducing the ones that correspond to frames of fragments that have not been selected to any step of the game, resulting in a new set of frame-level weights $\boldsymbol{s'}$. For the i^{th} step, these action-weighting factors ($Aw F$) for promoting the selected fragments are computed as follows:

$$Aw F_i = \frac{N-(i-1)}{M-(i-1)} + 1,\ i \in [i, N] \tag{1}$$

The reasoning behind the computation of the action-weighting factors is that the model needs to pay more attention to the first-selected fragments, thus the action-weighting factor in step i is larger than the one in step $i+1$.

The reduction factor (RF) is applied to the non-selected fragments only once at the end of the game, and is computed as follows:

$$RF = (M-N)/M \tag{2}$$

A **Critic** (fully connected network), who is also involved in the "N-picks" game and in every step i (with $1 \leq i \leq N$) of this game: i) gets the current state $\boldsymbol{f_i}$ (generated either at the beginning of the game by the State Generator, or as a result of the Actor's choices in every step of the game) and ii) computes a value v_i about this state, as an assessment of the Actor's choice.

A **Fragment Selector** (matrix multiplication operator), which uses the updated frame-level scores after each step of the game **s'**, that carry information about the Actor's preferences with regards to the most important (key) fragments of the video, to assign scores to the compressed features of the video frames ($\boldsymbol{X'} = \{\boldsymbol{x'_t}\}_{t=1}^{T}$) and produce a weighted version of them ($\boldsymbol{W} = \{\boldsymbol{w_t}\}_{t=1}^{T}$).

A **Variational Auto-Encoder** (LSTMs), which tries to discover the underlying structure of the weighted data after the Actor's choices and reconstruct the original video frames ($\hat{\boldsymbol{X}} = \{\hat{\boldsymbol{x}}_{\boldsymbol{t}}\}_{t=1}^{T}$). The goal of this encoding-decoding process is

to minimize the reconstruction error and produce a representation of the original video that fools the Discriminator.

A **Discriminator** (LSTM), which forms the AC environment and in every step i (with $1 \leq i \leq N$) of this game: i) gets the compressed feature vectors of the original video **X'** and the feature vectors of its reconstructed version, based on the Actor's choices and the subsequent encoding-decoding process, $\hat{\mathbf{X}}$, ii) defines a new latent representation for each of the aforementioned versions of the video, iii) computes a reconstruction loss (scalar value) based on the proximity of these representations, and iv) returns a reward to the Critic that is calculated as follows:

$$r_i = 1 - L_{recon}, \;\; r_i \in \mathbb{R}, \; i \in [i, N] \tag{3}$$

When the action sampled by the Actor leads to the selection of an already selected fragment, then the returned reward equals to zero to penalize the fragment's re-selection.

C. Learning Objectives and Pipeline

1) Learning Objectives: The learning objectives for training the State Generator, Encoder, Decoder and Discriminator of the proposed AC-SUM-GAN architecture include: a regularization loss ($L_{sparsity}$), a prior loss (L_{prior}), a reconstruction loss (L_{recon}), the "original" (L_{ORIG}) and "summary" (L_{SUM}) losses, and the generator loss (L_{GEN}). For sake of space we provide a short explanation of these losses and refer the reader to [6], [9] for a more detailed description. Then, we present the losses of the newly introduced components in the architecture.

$L_{sparsity}$ aims to force the State Generator to produce a sparse and diverse set of scores based on a regularization factor σ. L_{prior} measures how much information is lost when using the Encoder's latent space to represent the VAE's prior distribution. L_{recon} estimates the distance between the original and the reconstructed feature vectors. L_{ORIG} and L_{SUM} relate to a label-based training approach (labels "1" and "0" denote the original and the reconstructed feature vectors for the adversarial part of our method) and used to train the Discriminator; L_{ORIG} is used to minimize the difference between the computed probability and the "video" label when the Discriminator gets the original video, and L_{SUM} is used to minimize the difference between the computed probability and the "summary" label when the Discriminator gets the summary-based reconstructed video. Finally, L_{GEN} is used to minimize the difference between the probability computed by the Discriminator when the latter is fed with the reconstructed video and the "video" label, thus forcing the Generator to reconstruct a video that is indistinguishable from the original.

With regards to the training of the introduced AC model, the Actor uses the received feedback from the Critic after each step of the "N-picks" game, and aims to learn a policy that maximizes the probability of an important fragment to be used during the summary generation. This goal is captured by the following loss:

$$L_{actor} = -\frac{1}{N}\left(\sum_{i=1}^{N} ln\boldsymbol{c_i} \; \alpha_i + \delta \sum_{i=1}^{N} H(\boldsymbol{c_i})\right) \tag{4}$$

where $ln\boldsymbol{c_i}$ and $H(\boldsymbol{c_i})$ represent the logarithm and the entropy of the calculated probability density function $\boldsymbol{c_i}$ at each step of the game, α_i is the advantage that indicates how much better it is to take a specific action compared to the average action at the i^{th} state of the game, and δ is an entropy regularization coefficient. The advantage is defined as the difference between the returns z_i and the values v_i computed by the critic:

$$\alpha_i = z_i - v_i, \;\; i \in [1, N] \tag{5}$$

The return is the discounted cumulative reward of all steps and is computed by the following formula:

$$z_i = \sum_{k=i}^{N} \gamma^{k-i} r_i \tag{6}$$

where r_i is the Discriminator's reward at the i^{th} step of the game, and γ is the discount factor that shows how important future rewards are to the current state ($\gamma \in \mathbb{R}$, $0 \leq \gamma \leq 1$).

Finally, the Critic tries to learn how to evaluate the Actor's choice at the i^{th} step of the game by computing a scalar value v_i. Its training is based on the following loss:

$$L_{critic} = \frac{1}{N} \sum_{i=1}^{N} \alpha_i^2 \tag{7}$$

2) Learning Pipeline: The learning process is comprised of four distinct steps (four pairs of forward and backward passes), in each of which a different part of the AC-SUM-GAN architecture is trained (Figs. 3 and 4). Specifically, in the 1^{st} step, the algorithm performs a forward pass through the entire network, computes L_{prior} and L_{recon} and makes a backward pass to update the Encoder. In the 2^{nd} step, after a forward pass of the partially updated architecture, it computes the L_{recon} and L_{GEN} and uses their sum to update the Decoder. The 3^{rd} step is implemented in two sub-steps. In particular, a forward pass of the (once again) partially updated model leads to the creation of the reconstructed feature vectors $\hat{\boldsymbol{X}}$, which are then used for calculating L_{SUM}. Subsequently, the compressed feature vectors $\boldsymbol{X'}$ are fed to the Discriminator and L_{ORIG} is calculated. The gradients computed from the losses after two individual backward passes are accumulated and used to update the Discriminator and the linear compression layer that affects the compressed feature vectors.

The training of the remaining components, namely the State Generator, the Actor and the Critic is carried out in the 4^{th} step of this incremental process, as depicted in Fig. 4. More precisely, the original feature vectors $\boldsymbol{X}$ pass through the first three components of the partially updated model and produce the initial state ($\boldsymbol{f_1} = \{f_j\}_{j=1}^{M}$) of the action-state space. The latter is given as input to the Actor and Critic which then play the "N-picks" game. In every step i of this game (this iterative process is denoted by the "For loop" and the dashed-line bounding box in Fig. 4) the Critic computes a scalar value v_i to assess the current state, while the Actor takes an action by generating and sampling the distribution $\boldsymbol{c_i}$. This action affects the computed frame-level weights $\boldsymbol{s}$, resulting in $\boldsymbol{s'}$. As explained in Section III-B, these scores pass through the remaining components of the architecture that also take part in the game during this 4^{th} step. The reconstructed video

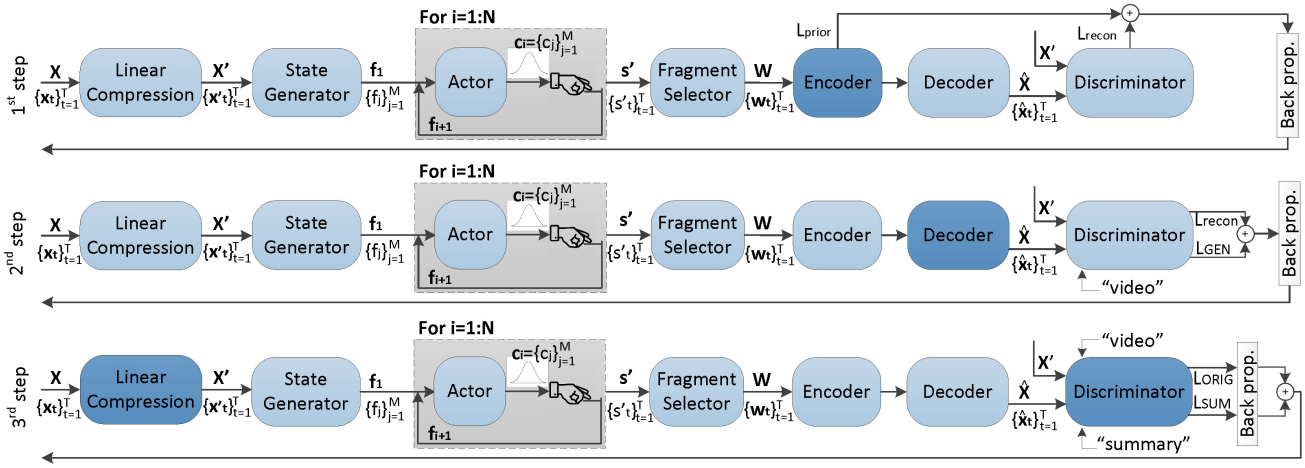


Fig. 3. The first three steps of the incremental training procedure. Dark-coloured boxes denote the parts updated in each step.

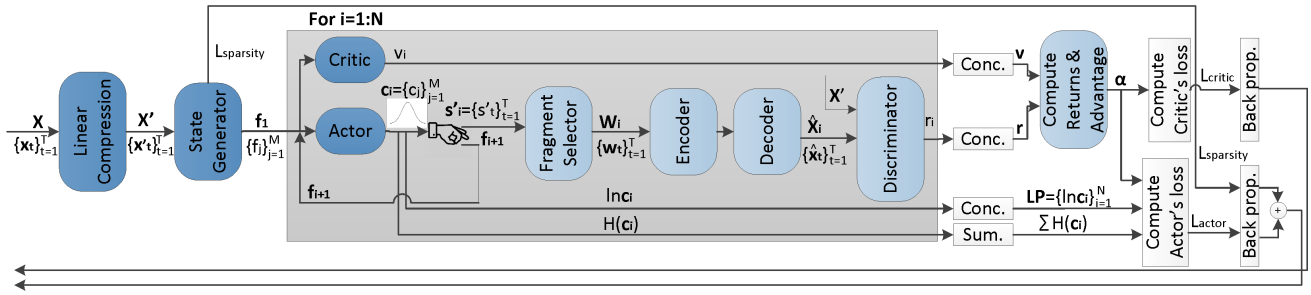


Fig. 4. The 4^{th} step of the incremental training procedure. Dark-coloured boxes denote the parts updated in this step.

is finally assessed by the Discriminator, which computes a reward r_i at each step of the game.

At the end of the game, the architecture produces the vectors $\boldsymbol{v} = \{v_i\}_{i=1}^{N}$, $\boldsymbol{r} = \{r_i\}_{i=1}^{N}$, $\boldsymbol{LP} = \{lnc_i\}_{i=1}^{N}$, and the scalar value $En = \sum_{i=1}^{N} H(\boldsymbol{c_i})$, whose elements have been previously described. The former two are used to compute the maximum expected returns and subsequently the advantage of taking a specific action compared to the average, general action at each given state. The computed advantages contribute to the training of the Critic. The training of the Actor is performed simultaneously with the training of the State Generator in a stepwise manner, similar to the Discriminator's training process. It uses the computed advantages $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^{N}$, $\boldsymbol{LP}$ and En values to form the L_{actor} and train the Actor, and the $L_{sparsity}$ that trains the State Generator. In this update step, the linear compression layer is also trained.

The added complexity with regards to [9] is the introduction of the AC model (composed of fully connected networks) for key-fragment selection and the design of a training process that uses the Discriminator's feedback as a reward. However, as shown in Fig. 5, the applied stepwise learning process allows all the different components to be trained effectively, and the AC-SUM-GAN model gets higher rewards as the training proceeds (see the bottom-right sub-figure of Fig. 5).

IV. Evaluation Setup

A. Datasets and Evaluation Protocols

1) Datasets: The performance of our unsupervised AC-SUM-GAN method is evaluated on the SumMe [1] and TVSum [2] datasets. SumMe includes 25 videos of 1 to 6 minutes duration, with diverse video contents, captured from both first-person and third-person view. Each video has been annotated by $15-18$ users in the form of key-fragments, and thus is associated to multiple fragment-level user summaries. Apart from that, a single ground-truth summary is provided for supervised training, computed by averaging the key-fragment summaries per frame. TVSum consists of 50 videos of 1 to 11 minutes duration, containing video content from 10 categories of the TRECVid MED dataset. The TVSum videos have been annotated by 20 users in the form of frame-level importance scores (ranging from 1 to 5), while a single ground-truth summary for each video (computed by averaging all users' scores for that video on a frame-basis) is also available.

2) Evaluation Metrics and Protocol: The most commonly used evaluation protocol is the key-fragment-based approach proposed in [4]. According to this protocol, the similarity between a machine-generated and a user-defined ground-truth summary is represented by expressing their overlap using the F-Score (as percentage). This protocol can be directly applied on the user summaries of the SumMe dataset, while its application on TVSum requires to transform the original frame-level annotations into key-fragment-based summaries [4]. Finally, for a given video and a machine-generated summary, this protocol matches the latter against all the available user summaries for this video and computes a set of F-Scores. For TVSum the final outcome occurs by averaging the computed F-Scores, while for SumMe this output corresponds to the maximum value among the computed F-Scores (as suggested in [50]). A few works ([9], [10], [20], [25], [30], [31]) follow

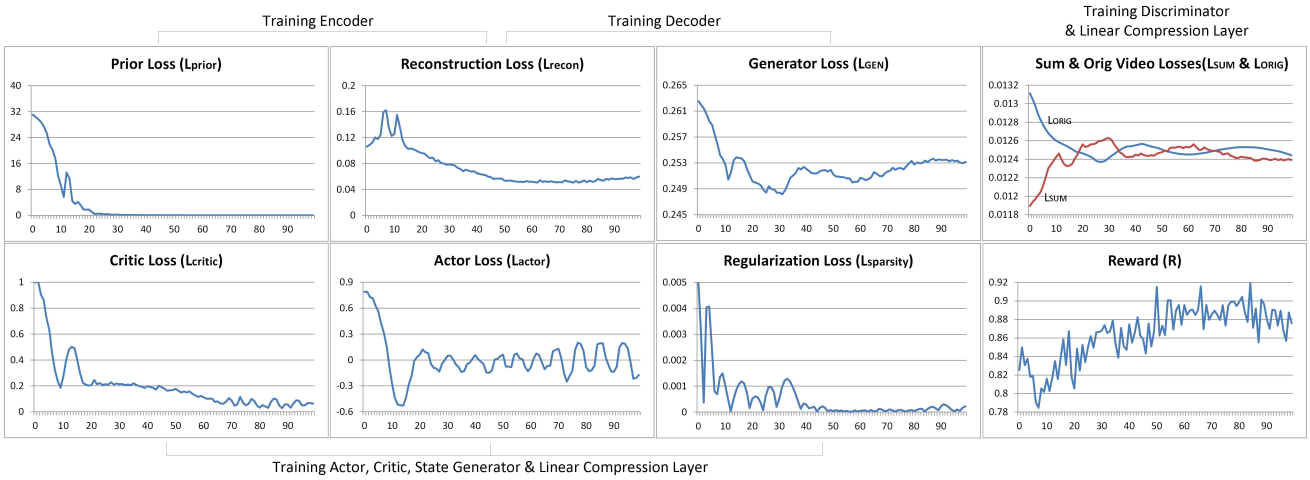


Fig. 5. Loss and reward curves for the proposed model. The horizontal axis in all plots indicates the epoch number. These curves indicate the successful training of Encoder, Decoder, Actor, Critic, State Generator and Discriminator, and the model's ability to get higher rewards as the training proceeds.

a slight variation of this evaluation protocol, which relies on the use of the single ground-truth summary that is available for each video of the above mentioned datasets.

In this work we adopt both the evaluation approach proposed in [4], and its aforementioned variation, to allow comparison with as many literature works on summarization as possible. Concerning the split of data for training and testing, we again follow the established approach (e.g., [4] and most literature works) of using 80% of the videos of each dataset for training and the remaining 20% for testing; and, we run experiments on five different randomly-generated splits for each dataset and report the average performance.

B. Implementation Details

As in most SoA methods, videos were downsampled to 2 fps. Then M, the number of non-overlapping and temporally equal video fragments, is dictated by the shortest video in the dataset, which in our case is 60 frames. So, $M = 60$ is the most fine-grained video representation possible. This hyper-parameter is the same for all videos so that the AC action-state space is of fixed dimensionality, as required for training the AC model. The duration d of each video fragment equals to the number of frames of a video divided by M. The target summary length must not exceed 15% of the original video duration, a convention adopted by most video summarization approaches (see Section III-B), thus also adopted in this work to allow for direct comparisons. With regards to the number of steps N, given the target summary length, this is calculated as $N = 15\% \cdot M = 9$. Deep representations of frames were obtained by taking the output of the pool5 layer of GoogleNet [51] trained on ImageNet (similar deep features are used in most SoA works). The linear compression layer reduces the size of feature vectors from 1024 to 512. The State Generator, Encoder, Decoder and Discriminator components are composed of 2-layer LSTMs with 512 hidden units, while the State Generator's LSTM is a bi-directional one. Actor and Critic consist of 4 and 5 fully connected layers respectively (see Fig. 6). The output of the last layer of the Actor is fed to a softmax layer, to form a categorical distribution of probabilities. The output of the last layer of the Critic is a scalar value between 0 and 1. The value of the discount factor γ is set to 0.99 in order to assign high importance to future rewards. The value of the entropy regularization coefficient δ is set to 0.1, following the example of other publicly-available implementations of the Actor-Critic model.[1] Finally, the AC-SUM-GAN model is trained in a full-batch mode (i.e., batch size is equal to the number of training samples) using the Adam optimizer. The learning rate for all components but the Discriminator is 10^{-4} and for the latter one is 10^{-5}. Training stops after a maximum number of epochs (100 in our case), and a well-trained model is selected according to a designed criterion which targets the maximization of the received rewards and the simultaneous minimization of the Actor's loss (a study on different criteria for the model selection is presented in Section V-A). To promote reproducibility of our reportings, the PyTorch implementation of the AC-SUM-GAN model is publicly-available at: https://github.com/e-apostolidis/AC-SUM-GAN.

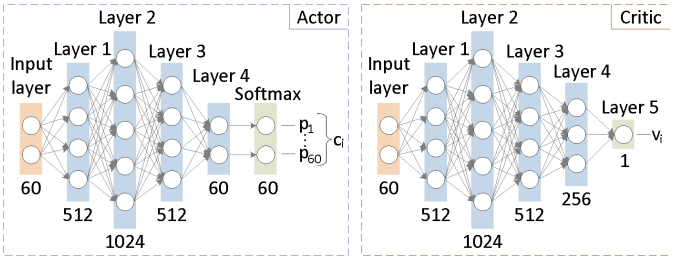


Fig. 6. The architecture of Actor and Critic models. The values below each layer's sketch represent the size of the layer (number of nodes).

V. Experimental Results

A. Selecting the Trained Model

We start our experimentation by studying different criteria for selecting a well-trained model after the end of the

[1] https://github.com/dennybritz/reinforcement-learning/tree/master/PolicyGradient

TABLE II

PERFORMANCE COMPARISON FOR DIFFERENT MODEL SELECTION CRITERIA. VALUES REPRESENT F-SCORE (%)

Criterion / Dataset	Reward	Reward & Actor loss	Recon. loss	Recon. & Sparsity loss	Reward & Recon. loss
SumMe	49.0	**50.8**	50.1	49.8	49.0
TVSum	60.5	60.6	60.7	**60.8**	60.0

unsupervised training process. In particular, we evaluate the performance of the introduced AC-SUM-GAN architecture when the trained model is selected based on the training set only and according to:

- The maximization of the overall received reward, computed as the mean of the received rewards r_i after each step of the "N-picks" game (so $i \in [1, N]$) that guide the training of the Actor-Critic model (the reward is a typical factor for early stopping when training relies on reinforcement learning; such a criterion is used in [5]).
- The maximization of the overall received reward and the simultaneous minimization of the Actor's loss L_{actor}, which is the main component of the AC-SUM-GAN model that is involved in the key-fragment selection process during the inference stage.
- The minimization of the reconstruction loss L_{recon} that signifies a maximum alignment between the original and the summary-based reconstructed video, and thus a representative summary.
- The simultaneous minimization of the reconstruction L_{recon} and sparsity losses $L_{sparsity}$; the latter is used (in combination with L_{actor}) for training the model's components used at the inference stage (i.e., the linear compression layer, the State Generator and the Actor).
- The maximization of the overall received reward and the simultaneous minimization of the reconstruction loss L_{recon}, that both indicate maximum similarity between the original and the summary-based reconstructed video, and thus a representative summary.

Driven by the remarks in [9] about the impact of the regularization factor σ on the summarization performance, we consider several values for this parameter (i.e., σ ranges in $[0.1, 1]$ with a step equal to 0.1). Instead of manually choosing a value, the best value for σ is also selected based on the used criterion for model selection. So, this criterion is responsible for selecting a well-trained model by indicating both the training epoch and the value of the regularization factor σ.

The results reported in Table II show that the impact of the employed criterion is much more pronounced on the SumMe dataset, whereas on the TVSum dataset different criteria lead to much smaller variation. Based on these results, we select and use in all subsequent experiments as criterion for model selection, the maximization of the overall received reward and the simultaneous minimization of the Actor's loss, which leads to the highest performance on SumMe and a near-optimal performance on TVSum.

TABLE III

COMPARISON WITH DIFFERENT UNSUPERVISED VIDEO SUMMARIZATION APPROACHES, ON SUMME AND TVSUM. F1 DENOTES F-SCORE (%) AND RNK DENOTES THE RANKING OF THE COMPARED METHODS

	SumMe		TVSum		Avg
	F1	Rnk	F1	Rnk	Rnk
Random summary	40.2	11	54.4	9	10
Online Motion-AE [45]	37.7	12	51.5	11	11.5
SUM-FCN$_{unsup}$ [27]	41.5	9	52.7	10	9.5
DR-DSN [5]	41.4	10	57.6	6	8
EDSN [42]	42.6	8	57.3	7	7.5
UnpairedVSN [41]	47.5	5	55.6	8	6.5
PCDL [44]	42.7	7	58.4	4	5.5
ACGAN [40]	46.0	6	58.5	3	4.5
SUM-GAN-sl [6]	47.8	4	58.4	4	4
SUM-GAN-AAE [7]	48.9	3	58.3	5	4
CSNet [8]	**51.3**	1	58.8	2	**1.5**
AC-SUM-GAN (Ours)	50.8	2	**60.6**	1	**1.5**

B. Evaluation Results and Comparisons

The performance of AC-SUM-GAN is initially compared against a random summarizer and a set of SoA unsupervised video summarization methods, on the SumMe and TVSum datasets. To estimate the performance of a random summarizer, importance scores for each frame are randomly assigned based on a uniform distribution of probabilities. The corresponding fragment-level scores are then used to form video summaries using the Knapsack algorithm and a length budget of maximum 15% of video duration. Random summarization is performed 100 times for each video, and the overall average score is reported. The results in Table III show that: i) the use of GANs for unsupervised learning of the video summarization task is a good choice, as the five top-performing methods (AC-SUM-GAN, CSNet, SUM-GAN-AAE, SUM-GAN-sl, ACGAN) rely on this learning framework; ii) algorithms that use reinforcement learning and tailored reward functions (DR-DSN, EDSN) are less competitive than the GAN-based approaches, especially on SumMe; iii) a few methods (placed at the top of the table) perform approximately equally to the random summarizer in at least one of the used datasets; finally, iv) the top-performing methods (AC-SUM-GAN, CSNet) try to tackle the limitation of the LSTM-based models that relates to the low variance of the predicted importance scores for the video frames. Concerning the top-performing methods, we see that AC-SUM-GAN is the best on TVSum and the second best on SumMe, while the opposite is observed for CSNet; so, practically we have a tie between these two methods. The competitive performance of CSNet is mainly affected by the use of a tailored variance loss function which aims to increase the variance of the estimated frame-level importance scores. In our AC-SUM-GAN method the boost in performance is gained by the use of a trained AC model that uses the Discriminator's feedback to learn a policy for key-fragment selection.

Our unsupervised AC-SUM-GAN model is also compared with SoA supervised video summarization approaches, despite the fact that this is a rather unfair comparison for our method. The data presented in Table IV shows that: i) once again a few methods (placed at the top of the table) exhibit random performance in at least one of the used datasets; ii) a number

of summarization techniques (Tessellation, MAVS) that exhibit high performance on one dataset perform very poorly on the other; iii) the proposed unsupervised AC-SUM-GAN model performs consistently well on both datasets and, based on the average ranking after considering both datasets, is the 3^{rd} top-performing method among a large set of SoA supervised techniques; finally, iv) the three best-performing approaches utilize tailored attention mechanisms (VASNet, H-MAN) or memory networks (SMN) to capture variable- and long-range temporal dependencies respectively, and we attribute their good performance on these mechanisms.

In addition, for fair comparison with video summarization approaches that utilize the single ground-truth summary for evaluation (the variation of the evaluation protocol of [4], as discussed in Section IV-A), we also assess the performance of AC-SUM-GAN with this protocol. In Table V the performance of the AC-SUM-GAN method is compared with the performance of the few supervised and unsupervised methods that adopt the aforementioned evaluation protocol. On SumMe, AC-SUM-GAN is by far the best-performing method, surpassing the second best approach (the supervised Ptr-Net algorithm) by more than 14 percentage points. On TVSum, AC-SUM-GAN is again the top-performing method. In addition, the introduction of the Actor-Critic model for key-fragment selection leads to a noticeable performance improvement compared to the original SUM-GAN model (by more than 22 percentage points on SumMe and by 14 percentage points on TVSum) that was the basis for our developments. Overall, the proposed unsupervised AC-SUM-GAN method performs consistently well on both datasets and is the best among the examined supervised and unsupervised algorithms.

C. Ablation Study

To assess the contribution of each of the major components of our model, we conduct an ablation study. This study involves the following variants of the AC-SUM-GAN model:

- **AC-SUM-GAN w/o VAE**. This variant excludes the Variational Auto-Encoder, and the weighted feature vectors at the output of the Fragment Selector are directly forwarded to the Discriminator (i.e., $\hat{\mathbf{X}} = \mathbf{W}$). Therefore, the incremental training of this variant involves only the 3^{rd} and 4^{th} step of the entire process (see Fig. 3 and 4).
- **AC-SUM-GAN w/o Discriminator**. This variant leaves out the Discriminator. Hence, the model is not trained under an adversarial manner and the similarity between the original and summary-based reconstructed version of the video (expressed by the reconstruction loss) is estimated through the direct comparison of the corresponding feature vectors. As a consequence, the 3^{rd} step of the incremental training process of Fig. 3 is omitted.
- **AC-SUM-GAN w/o Actor-Critic**. This variant does not contain the Actor-Critic model and the State Generator's function $F(s)$ that is essential only for training the Actor-Critic model. Consequently, the 4^{th} step of the applied training process (see Fig. 4) updates only the State Generator and the linear compression layer using the sum of $L_{sparsity}$ and L_{recon}.

[2]This literature work uses a different evaluation protocol; for this reason we do not present this result here.

TABLE IV

COMPARISON OF OUR UNSUPERVISED METHOD WITH SUPERVISED VIDEO SUMMARIZATION APPROACHES ON SUMME AND TVSUM. F1 DENOTES F-SCORE (%) AND RNK DENOTES THE RANKING OF THE COMPARED METHODS

	SumMe		TVSum		Avg
	F1	Rnk	F1	Rnk	Rnk
Random summary	40.2	26	54.4	22	24
vsLSTM [4]	37.6	29	54.2	23	26
dppLSTM [4]	38.6	28	54.7	21	24.5
SASUM [20]	40.6	24	53.9	24	24
APDVS [17]	41.2	21	51.3	25	23
ActionRanking [36]	40.1	27	56.3	19	23
ESS-VS [13]	40.9	23	–	–	23
H-RNN [21]	41.1	22	57.7	16	19
vsLSTM+Att [23]	43.2	18	– [2]	–	18
DSSE [19]	–	–	57.0	17	17
DR-DSN$_{sup}$ [5]	42.1	19	58.1	14	16.5
dppLSTM+Att [23]	43.8	15	– [2]	–	15
WS-HRL [39]	43.6	17	58.4	12	14.5
UnpairedVSN$_{psup}$ [41]	48.0	7	56.1	20	13.5
SUM-FCN [27]	47.5	9	56.8	18	13.5
SF-CVS [37]	46.0	12	58.0	15	13.5
SASUM$_{fullysup}$ [20]	45.3	13	58.2	13	13
MAVS [28]	40.3	25	**66.8**	1	13
CRSum [34]	47.3	10	58.0	15	12.5
PCDL$_{sup}$ [44]	43.7	16	59.2	9	12.5
Tessellation [12]	41.4	20	64.1	3	11.5
HSA-RNN [22]	44.1	14	59.8	7	10.5
DQSN [16]	–	–	58.6	10	10
ACGAN$_{sup}$ [40]	47.2	11	59.4	8	9.5
SUM-DeepLab [27]	48.8	5	58.4	12	8.5
CSNet$_{sup}$ [8]	48.6	6	58.5	11	8.5
SMLD [35]	47.6	8	61.0	5	6.5
H-MAN [26]	51.8	2	60.4	7	4.5
VASNet [24]	49.7	4	61.4	4	4
SMN [29]	**58.3**	1	64.5	2	**1.5**
AC-SUM-GAN (Ours)	50.8	3	60.6	6	4.5

TABLE V

COMPARISON OF OUR UNSUPERVISED METHOD WITH OTHER VIDEO SUMMARIZATION APPROACHES ON SUMME AND TVSUM, USING A SINGLE GROUND-TRUTH SUMMARY FOR EACH VIDEO. UNSUPERVISED METHODS ARE MARKED WITH *. F1 DENOTES F-SCORE (%) AND RNK DENOTES THE RANKING OF THE COMPARED METHODS

	SumMe		TVSum		Avg
	F1	Rnk	F1	Rnk	Rnk
Random Summary	40.2	8	54.4	8	8
*SUM-GAN [9]	38.7	9	50.8	9	9
SUM-GAN$_{sup}$ [9]	41.7	7	56.3	7	7
*Cycle-SUM [10]	41.9	6	57.6	6	6
A-AVS [25]	43.9	5	59.4	4	4.5
DTR-GAN [30]	44.6	3	59.1	5	4
M-AVS [25]	44.4	4	61.0	3	3.5
Ptr-Net [31]	46.2	2	63.6	2	2
*AC-SUM-GAN (Ours)	**60.7**	1	**64.8**	1	**1**

To eliminate the impact of the model selection criterion, in this set of experiments we consider a fixed σ value equal to 0.5 (which is the median of the σ values considered in our experiments) and manually select the best trained model according to its performance on the test set (thus, a performance higher to the reported one in Tables III and IV can be recorded). Once again, we run this experiment on the same group of five randomly-created data splits and we

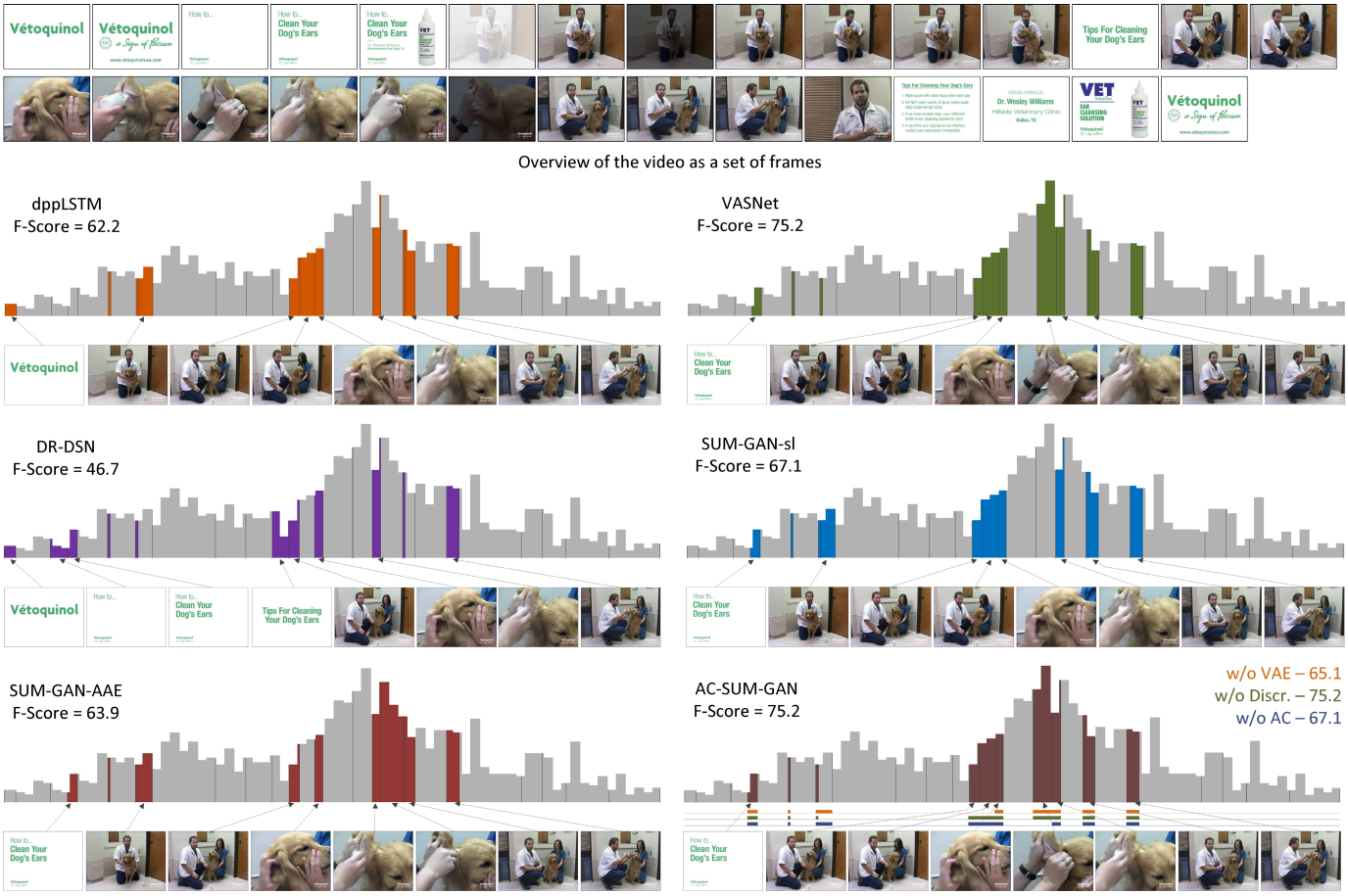


Fig. 7. A key-frame-based overview (using one key-frame per shot), and example summaries of six summarization methods on video #15 of the TVSum dataset (the first two methods, dppLSTM and VASNet, are supervised, while the rest are unsupervised). For AC-SUM-GAN, we also illustrate with coloured horizontal line segments under the corresponding bar-chart, the result of each of the three variations of it discussed in the ablation study (Section V-C).

TABLE VI

ABLATION STUDY BASED ON THE PERFORMANCE (F-SCORE (%)) OF THREE VARIATIONS OF THE PROPOSED MODEL, ON SUMME AND TVSUM

	SumMe	TVSum
AC-SUM-GAN w/o VAE	53.0	61.1
AC-SUM-GAN w/o Discriminator	53.3	60.7
AC-SUM-GAN w/o Actor-Critic	50.4	60.7
AC-SUM-GAN	**54.5**	**61.4**

report the average performance. The results in Table VI show that the introduction of the Actor-Critic model has a clearly positive impact on the summarization performance on both datasets, which is more pronounced on SumMe. Moreover, the other two major components of the proposed architecture, i.e., the Variational Auto-Encoder and the Discriminator, are also shown to have a positive impact on performance.

In order to investigate what is the computational complexity of embedding an AC model into GAN-based summarization architectures (such as the SUM-GAN model and its existing variations), we measured the training and inference times for AC-SUM-GAN against its variation without AC. Results averaged over five data splits of the SumMe and TVSum datasets show that the training time is increased by 55% - this is expected given the additional parameters that need to be learned; however, there is no noticeable difference at the inference stage - in both cases, video summarization takes less than 0.2 seconds.

D. Qualitative Analysis - A Summarization Example

In addition to the above reported findings, we illustrate the quality of the produced summaries by the proposed AC-SUM-GAN method with an example. For this, we use video #15 of the TVSum dataset (titled "How to Clean Your Dog's Ears - Vetoquinol USA") that is used for the same purpose in a few other SoA works (e.g., [4], [8]–[10], [39], [40]), and we compare the performance of the AC-SUM-GAN method against five other summarization methods with publicly-available implementations (these methods are, to our knowledge, the only ones for which implementations are publicly available). Fig. 7 gives an overview of the video after selecting one frame per shot (shot segmentation performed using the KTS method) and presents the results for the examined techniques. In each case, the gray bars denote the averaged human-annotated importance scores for the frames of the video, the black vertical lines within these bars correspond to the shot boundaries, and the coloured bars indicate the selected key-shots for creating the summary. Moreover, for each method we provide an illustration of the generated

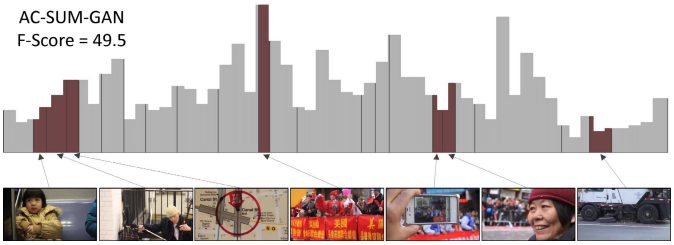


Fig. 8. Example of a video summary with limited overlap with the ground-truth annotations.

summary by selecting one representative key-frame from each one of the major key-shots of the summary. These results show that the proposed unsupervised AC-SUM-GAN method generates the exact same summary with the VASNet algorithm, which is one of the best-performing supervised summarization approaches on TVSum. And the superiority of these two algorithms is proven also in terms of F-Score (see values plotted under each method's name). The generated summary focuses on the main event of the video (i.e., the cleaning of the dog's ears), but it also contains shots with diverse visual content from other parts of the video. In this way, it provides a comprehensive presentation of the entire story, with a special focus on its main event. Regarding the other techniques, the SUM-GAN-AAE algorithm also selects some fragments of top importance, ending up to a visually similar result with AC-SUM-GAN and VASNet (the difference in terms of F-Score is due to the imperfection of the KTS method, which erroneously splits one shot in more shots; and, in this example, such a fragment that is visually similar with the best selection ended up in the summary). The three remaining methods focus less on the main event, with DR-DSN losing the point of the video and choosing many frames that mainly contain graphics.

To examine the impact of each of the main components of the AC-SUM-GAN architecture on the summarization outcome, at the bottom-right part of Fig. 7 we illustrate also the selected fragments by each different variation of the AC-SUM-GAN model. The coloured line segments right below the bar-chart show that the variation without the Discriminator produces the exact same summary with the AC-SUM-GAN method. The other two variations lead to different and slightly worse summaries. The model without the AC part misses the selection of the most important part of the video, while the model without the VAE also misses some important part of the main story by instead selecting a video part that is of lower importance according to the ground-truth annotations. These findings are consistent with the findings of the conducted ablation study and indicate the positive impact of the introduced AC model in the summarization performance.

Experimentation with other videos of the used datasets, showed that there are cases where the summaries created by our method have limited overlap with the ground-truth annotations. Indicatively, in Fig. 8 we show the ground-truth annotation (gray-coloured bars) and the selected fragments (brown-coloured bars) for video #26 of the TVSum dataset (titled "Chinese New Year Parade 2012 NY City Chinatown"). In this video the AC-SUM-GAN picks some parts from the beginning and end of the video, and misses some more important parts from the middle of the video that show the actual parade. This example demonstrates that video summarization is a difficult problem and further technological advancements are needed to fully meet the human expectations.

VI. Conclusion and Future Work

In this work we introduced a new formulation of the video summarization task, that tackles the selection of the most important parts of the video as a "visual sentence" generation process. The proposed method embeds an Actor-Critic model into a Generative Adversarial Network for unsupervised video summarization. The feedback of the Discriminator is used to train the Actor and Critic models through their participation in a fragment selection game. The designed training strategy allows the Critic to learn a value function and the Actor to learn a policy for key-fragment selection. The proposed model selection criterion, that relies on the optimization of core factors of the training process (i.e., the received reward and the loss function of the Actor), assists with the selection of proper values for the model's parameters. Experiments on two benchmark datasets placed the proposed method among the top-performing unsupervised video summarization algorithms, and indicated its competitiveness against the majority of SoA supervised approaches. The outcomes of the conducted ablation study pointed out the benefits of connecting an Actor-Critic model with a Generative Adversarial Network for unsupervised video summarization.

Future plans towards further advancing the AC-SUM-GAN method's performance include, first, investigating the merits of using a Soft Actor-Critic [52] that is capable of further discovering the action space by automatically defining a suitable value for the entropy regularization factor. Second, we will investigate the introduction of a chunk and stride network (such as the one in [8]) or the extension of the State Generator by a memory network (similar to [29]), to capture long-range dependencies and produce better fragment scoring, thus facilitating the Actor's training and leading to better choices during the key-fragment selection.

References

[1] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating Summaries from User Videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 505–520.

[2] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "TVSum: Summarizing Web videos using titles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5179–5187.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 766–782.

[5] K. Zhou and Y. Qiao, "Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 1–9.

[6] E. Apostolidis, A. I. Metsai, E. Adamantidou, V. Mezaris, and I. Patras, "A stepwise, label-based approach for improving the adversarial training in unsupervised video summarization," in *Proc. 1st Int. Workshop AI Smart TV Content Prod., Access Del. (AITV)*. New York, NY, USA: ACM, 2019, pp. 17–25.

[7] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Unsupervised video summarization via attention-driven adversarial learning," in *Multimedia Modeling*, Y. M. Ro, W.-H. Cheng, J. Kim, W.-T. Chu, P. Cui, J.-W. Choi, M.-C. Hu, and W. De Neve, Eds. Cham, Switzerland: Springer, 2020, pp. 492–504.

[8] Y. Jung, D. Cho, D. Kim, S. Woo, and I. S. Kweon, "Discriminative feature learning for unsupervised video summarization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8537–8544.

[9] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial LSTM networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2982–2991.

[10] L. Yuan, F. E. H. Tay, P. Li, L. Zhou, and J. Feng, "Cycle-SUM: Cycle-consistent adversarial LSTM networks for unsupervised video summarization," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2019, pp. 9143–9150.

[11] A. Goyal *et al.*, "ACtuAL: Actor-critic under adversarial learning," 2017, *arXiv:1711.04755*. [Online]. Available: http://arxiv.org/abs/1711.04755

[12] D. Kaufman, G. Levi, T. Hassner, and L. Wolf, "Temporal tessellation: A unified approach for video analysis," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 94–104.

[13] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1059–1067.

[14] R. Panda, A. Das, Z. Wu, J. Ernst, and A. K. Roy-Chowdhury, "Weakly supervised summarization of Web videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3677–3686.

[15] X. Song *et al.*, "Category driven deep recurrent neural network for video summarization," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–6.

[16] K. Zhou, T. Xiang, and A. Cavallaro, "Video summarisation by classification with deep reinforcement learning," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–13.

[17] J. Lei, Q. Luan, X. Song, X. Liu, D. Tao, and M. Song, "Action parsing-driven video summarization based on reinforcement learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 2126–2137, Jul. 2019.

[18] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya, "Video summarization using deep semantic features," in *Proc. 13th Asian Conf. Comput. Vis. (ACCV)*, 2016, pp. 361–377.

[19] Y. Yuan, T. Mei, P. Cui, and W. Zhu, "Video summarization by learning deep side semantic embedding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 226–237, Jan. 2019.

[20] H. Wei, B. Ni, Y. Yan, H. Yu, X. Yang, and C. Yao, "Video summarization via semantic attended networks," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 216–223.

[21] B. Zhao, X. Li, and X. Lu, "Hierarchical recurrent neural network for video summarization," in *Proc. ACM Multimedia Conf. (MM)*. New York, NY, USA: ACM, 2017, pp. 863–871.

[22] B. Zhao, X. Li, and X. Lu, "HSA-RNN: Hierarchical structure-adaptive RNN for video summarization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7405–7414.

[23] L. Lebron Casas and E. Koblents, "Video summarization with LSTM and deep attention models," in *MultiMedia Modeling*, I. Kompatsiaris, B. Huet, V. Mezaris, C. Gurrin, W.-H. Cheng, and S. Vrochidis, Eds. Cham, Switzerland: Springer, 2019, pp. 67–79.

[24] J. Fajtl, H. S. Sokeh, V. Argyriou, D. Monekosso, and P. Remagnino, "Summarizing videos with attention," in *Proc. Asian Conf. Comput. Vis. (ACCV) Workshops*, G. Carneiro and S. You, Eds. Cham, Switzerland: Springer, 2019, pp. 39–54.

[25] Z. Ji, K. Xiong, Y. Pang, and X. Li, "Video summarization with attention-based encoder–decoder networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1709–1717, Jun. 2020.

[26] Y.-T. Liu, Y.-J. Li, F.-E. Yang, S.-F. Chen, and Y.-C.-F. Wang, "Learning hierarchical self-attention for video summarization," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 3377–3381.

[27] M. Rochan, L. Ye, and Y. Wang, "Video summarization using fully convolutional sequence networks," in *Proc. Europ. Conf. Comput. Vis. (ECCV)*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 358–374.

[28] L. Feng, Z. Li, Z. Kuang, and W. Zhang, "Extractive video summarizer with memory augmented neural networks," in *Proc. ACM Multimedia Conf. Multimedia Conf. (MM)*. New York, NY, USA: ACM, 2018, pp. 976–983.

[29] J. Wang, W. Wang, Z. Wang, L. Wang, D. Feng, and T. Tan, "Stacked memory network for video summarization," in *Proc. 27th ACM Int. Conf. Multimedia*. New York, NY, USA: ACM, Oct. 2019, pp. 836–844.

[30] Y. Zhang, M. Kampffmeyer, X. Zhao, and M. Tan, "DTR-GAN: Dilated temporal relational adversarial network for video summarization," in *Proc. ACM Turing Celebration Conf. China (ACM TURC)*, Beijing, China. New York, NY, USA: ACM, 2019, pp. 89:1–89:6.

[31] T.-J. Fu, S.-H. Tai, and H.-T. Chen, "Attentive and adversarial learning for video summarization," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Waikoloa Village, HI, USA, Jan. 2019, pp. 1579–1587.

[32] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2015, pp. 2692–2700.

[33] S. Lal, S. Duggal, and I. Sreedevi, "Online video summarization: Predicting future to better summarize present," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 471–480.

[34] Y. Yuan, H. Li, and Q. Wang, "Spatiotemporal modeling for video summarization using convolutional recurrent neural network," *IEEE Access*, vol. 7, pp. 64676–64685, 2019.

[35] W.-T. Chu and Y.-H. Liu, "Spatiotemporal modeling and label distribution learning for video summarization," in *Proc. IEEE 21st Int. Workshop Multimedia Signal Process. (MMSP)*, Sep. 2019, pp. 1–6.

[36] M. Elfeki and A. Borji, "Video summarization via actionness ranking," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Waikoloa Village, HI, USA, Jan. 2019, pp. 754–763.

[37] C. Huang and H. Wang, "A novel key-frames selection framework for comprehensive video summarization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 577–589, Feb. 2020.

[38] Y. Huang, X. Cao, Q. Wang, B. Zhang, X. Zhen, and X. Li, "Long-short-term features for dynamic scene classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 4, pp. 1038–1047, Apr. 2019.

[39] Y. Chen, L. Tao, X. Wang, and T. Yamasaki, "Weakly supervised video summarization by hierarchical reinforcement learning," in *Proc. ACM Multimedia Asia*. New York, NY, USA: ACM, Dec. 2019, pp. 1–6.

[40] X. He *et al.*, "Unsupervised video summarization with attentive conditional generative adversarial networks," in *Proc. 27th ACM Int. Conf. Multimedia*. New York, NY, USA: ACM, Oct. 2019, pp. 2296–2304.

[41] M. Rochan and Y. Wang, "Video summarization by learning from unpaired data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7902–7911.

[42] N. Gonuguntla *et al.*, "Enhanced deep video summarization network," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2019, pp. 1–9.

[43] L. Wang *et al.*, "Temporal segment networks: Towards good practices for deep action recognition," in *Comput. Vis. (ECCV)*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 20–36.

[44] B. Zhao, X. Li, and X. Lu, "Property-constrained dual learning for video summarization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 3989–4000, Oct. 2020.

[45] Y. Zhang, X. Liang, D. Zhang, M. Tan, and E. P. Xing, "Unsupervised object-level video summarization with online motion auto-encoder," *Pattern Recognit. Lett.*, vol. 130, pp. 376–385, Feb. 2020.

[46] D. Pfau and O. Vinyals, "Connecting generative adversarial networks and actor-critic methods," in *Proc. NIPS Workshop Adversarial Training*, 2016, pp. 1–10.

[47] N. Savioli, "A hybrid approach between adversarial generative networks and actor-critic policy gradient for low rate high-resolution image compression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4321–4324.

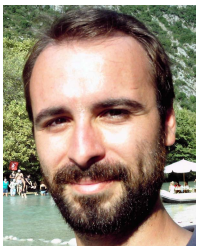
[48] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence generative adversarial nets with policy gradient," in *Proc. 31st AAAI Conf. Artif. Intell. (AAAI)*, 2017, pp. 2852–2858.

[49] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 540–555.

[50] M. Gygli, H. Grabner, and L. Van Gool, "Video summarization by learning submodular mixtures of objectives," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3090–3098.

[51] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[52] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 1–14.



Evlampios Apostolidis received the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2007, and the M.Sc. degree in information systems from the University of Macedonia, Thessaloniki, in 2011. He is currently pursuing the Ph.D. degree with the School of Electronic Engineering and Computer Science, Queen Mary University of London. His Diploma Thesis was on methods for digital watermarking of 3D TV content. For his Dissertation, he studied techniques for indexing multidimensional data. Since January 2012, he has been a Research Assistant with the Centre for Research and Technology Hellas, Information Technologies Institute. He has coauthored two journal articles, four book chapters, and more than 25 conference papers. His research interests include the areas of video analysis and understanding, with a particular focus on methods for video segmentation and summarization.

Eleni Adamantidou received the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2019. She graduated in the top 10% of her class (grade 9.01/10). Since March 2019, she has been working as a Research Assistant with the Centre for Research and Technology Hellas, Information Technologies Institute. She has coauthored four conference papers in the field of video summarization. She is particularly interested in deep learning methods for video analysis, video summarization, and natural language processing.

Alexandros I. Metsai received the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2017. For the needs of his diploma thesis, he developed a robotic agent capable of learning objects shown by a human through voice commands and hand gestures. Since September 2018, he has been working as a Research Assistant with the Centre for Research and Technology Hellas, Information Technologies Institute. He has coauthored four conference papers in the field of video summarization and one book chapter in the field of video forensics. His research interests include the area of deep learning for video analysis and summarization.

Vasileios Mezaris (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2001 and 2005, respectively. He is currently a Research Director with the Centre for Research and Technology Hellas, Information Technologies Institute. He has coauthored more than 40 journal articles, 20 book chapters, 170 conference papers, and three patents. His research interests include multimedia understanding and artificial intelligence, in particular, image and video analysis and annotation, machine learning and deep learning for multimedia understanding and big data analytics, multimedia indexing and retrieval, and applications of multimedia understanding and artificial intelligence. He serves as a Senior Area Editor for IEEE SIGNAL PROCESSING LETTERS and as an Associate Editor for IEEE TRANSACTIONS ON MULTIMEDIA.

Ioannis (Yiannis) Patras (Senior Member, IEEE) is currently a Professor in computer vision and human sensing with the School of Electronic Engineering and Computer Science, Queen Mary University of London. He has more than 200 publications in the most selective journals and conferences in the field of Computer Vision. His research interests include the areas of computer vision and human sensing using machine learning methodologies–this includes tracking and recognition of human actions and activity in multimedia, affect analysis, and multimodal analysis of human behavior. He is a member of the Visual Signal Processing and Communications Technical Committee (VSPC) of CAS society. He is/has been in the organizing committee of Multimedia Modeling 2021, ICMR 2019, IEEE Image, Video, and Multidimensional Signal Processing Symposium 2018, ACM Multimedia 2013, ICMR 2011, Face and Gesture Recognition 2008, BMVC 2009, and was the General Chair of WIAMIS 2009. He is an Associate Editor of the *Journal of Pattern Recognition*, *Computer Vision and Image Understanding*, and the Image and Vision Computing Journal, and the Area Chair of major Computer Vision conferences including, ICCV, ECCV, FG, and BMVC.