Does spatio-temporal information benefit the video summarization task?

Aashutosh Ganesh[1,*], Mirela Popa[1], Daan Odijk[2] and Nava Tintarev[1]

[1]Department of Advanced Computing Sciences, Maastricht University, the Netherlands
[2]RTL, Hilversum, the Netherlands

Abstract

An important aspect of summarizing videos is understanding the temporal context behind each part of the video to grasp what is and is not important. Video summarization models have in recent years modeled spatio-temporal relationships to represent this information. These models achieved state-of-the-art correlation scores on important benchmark datasets. However, what has not been reviewed is whether spatio-temporal relationships are even required to achieve state-of-the-art results. Previous work in activity recognition has found biases, by prioritizing static cues such as scenes or objects, over motion information. In this paper we inquire if similar spurious relationships might influence the task of video summarization. To do so, we analyse the role that temporal information plays on existing benchmark datasets. We first estimate a baseline with temporally invariant models to see how well such models rank on benchmark datasets (TVSum and SumMe). We then disrupt the temporal order of the videos to investigate the impact it has on existing state-of-the-art models. One of our findings is that the temporally invariant models achieve competitive correlation scores that are close to the human baselines on the TVSum dataset. We also demonstrate that existing models are not affected by temporal perturbations. Furthermore, with certain disruption strategies that shuffle fixed time segments, we can actually improve their correlation scores. With these results, we find that spatio-temporal relationship play a minor role and we raise the question whether these benchmarks adequately model the task of video summarization. Code available at: https://github.com/AashGan/TemporalPerturbSum

Keywords

Video Summarization, Trustworthiness, Self-attention, Temporal Disruptions

1. Introduction

A striking amount of short-form video content is created, hosted, and consumed within the online media landscape. Several platforms such as Tiktok, Youtube and Instagram promote these short snappy videos as they immediately capture the users' interest. These videos are often created by cutting a longer video into its best parts. However, the process of editing them into these bite-sized pieces is still time-intensive with significant potential for automation. One way to broach automatic editing is by using video summarization algorithms [1].

What makes a good video summary is however largely subjective and is dependent on the underlying media. Regardless of this subjectivity, a pivotal aspect in understanding what is pertinent for a video summary is the *temporal context* behind different parts of a video. The context in this case is the relationship one part of a video shares with other parts, as the preceding or succeeding frames or shots inform us as to what may be relevant. The information gained by learning the temporal context gives us vital cues regarding what should be included in a summary. Therefore, we may expect automatic video summarization algorithms by design to discover and exploit such relationships.

Based on this assumption, modern approaches to video summarization utilise Deep Neural Networks, which employ spatio-temporal relationships within the data [2, 3, 4, 5] to understand the temporal context within the videos. These approaches often estimate a frame-wise importance score, which indicates how likely is that a frame should be included in a summary. To evaluate the accuracy of their predictions, they measure how well these scores correlate with gold standard human labels provided

AEQUITAS 2024: Workshop on Fairness and Bias in AI | co-located with ECAI 2024, Santiago de Compostela, Spain
*Corresponding author.
Aashutosh.Ganesh@maastrichtuniversity.nl (A. Ganesh); m.popa@maastrichtuniversity.nl (M. Popa); Daan.Odjik@rtl.nl (D. Odijk); n.tintarev@maastrichtuniversity.nl (N. Tintarev)

by two important benchmark datasets: TVSum [6] and SumMe [7]. These datasets despite their small size, are a cornerstone within video summarization research due to their diverse set of videos and the inclusion of multiple human annotations per video to capture the subjectivity of video summarization.

An important question that can be raised regarding these benchmark datasets is the role spatio-temporal relationship play within them. Despite the state-of-the-art Kendall rank correlation coefficient scores achieved by these spatio-temporal models, it has been observed by Otani et al [1] that a simple multi-ranking model which does not account for temporal relationships by design, also achieved competitive scores. This casts a doubt regarding the trustworthiness of the aforementioned models to capture temporal context.

A similar question has also been posed within activity recognition research [8, 9]. Some of the relevant works showcase how activity recognition models could effectively ignore the temporal component of the video, while still accurately predicting the activity. In some cases, this could be due to a "representation bias" within the dataset. This bias, as illustrated by Li et al. [9], is where a dataset may favour a certain data representation, influencing the model to learn spurious relationships. For instance, videos could be collected for an activity such as playing football, may have associated videos from a specific field. This type of intrinsic correlation might be captured by the DNN model, which recognizes correctly the activity class by detecting the field instead of the football related motion. This may give us the false impression that an activity recognition model should simply recognize the salient objects or scenes. This also poses a significant challenge to the trustworthiness of such systems, since it may use unexpected and spurious relationships for predicting a certain activity class.

An investigation of the role temporal dependencies play within popular video summarization benchmark datasets is important in order to evaluate the trustworthiness of these methods. If a temporally invariant model achieves a performance that is on par with its temporally dependent counterparts, then this may highlight an issue within the benchmarks used to compare them. To analyse this hypothesis, experiments which manipulate the temporal order of a video will indicate whether these dependencies are truly needed. Furthermore, interesting insights could be formulated with respect to the amount of data needed to capture various temporal relationships. These observations highlight the growing need for scrutiny over benchmark datasets, by critically assessing whether the underlying data distribution is adequate to model the intended types of relationships, leading to increasing trust in AI systems.

Therefore in this work we investigate the role spatio-temporal dependencies play on the video summarization benchmark datasets. We achieve this goal by disrupting the temporal order of the videos. We first establish a baseline performance using temporally invariant models on TVSum and SumMe datasets. On this baseline, we investigate the effect that temporal disruptions have on models which utilise spatio-temporal relationships. We take inspiration from time series data augmentation [10] to design different temporal disruptions introduced at different timescales, including low-level, intermediate and global levels, represented by frames and shots. Through these experiments, we investigate the role played by the temporal context in video summarization on two important benchmark datasets.

To summarise, in this paper we make the following contributions:

- We demonstrate that models which do not utilise temporal context can still achieve close to the state-of-the-art correlation scores on the TVSum and SumMe datasets.
- We highlight that the introduction of temporal disruptions had a limited effect on the performance of video summarization models on the two considered benchmark datasets. Moreover, we also prove that the analyzed temporal disruptions in some cases even improved the models' performance, underscoring the role played by spatio-temporal relationships.
- Finally, we trace back the aforementioned results to the design of the datasets, the architecture and the evaluation stage. Our results indicate the need to address the identified limitations, when designing future benchmarks in video summarization, by properly evaluating the expected contributions at various temporal levels.

2. Related Work

Video summarization, as defined by Apostolidis et al. [11], is the task of retrieving a "synopsis" of a video by selecting the fewest and most pertinent parts of the video. Recent work favours the summaries creation in the form of video skims, due to user preferences [11]. The primary datasets which serve as benchmarks for the state-of-the-art are: TVSum [6] and SumMe [7]. Chhara et al. [12] is among the first works that introduces and evaluates the concept of "fairness" within video summarization. In this context fairness addresses equal representations of individuals and protected groups within a final summary. In our work, we evaluate whether spatio-temporal relationships play a role or if these benchmarks may be statically biased.

2.1. Supervised Video Summarization with spatio-temporal models

Among the earliest examples which utilised neural networks and spatio-temporal modeling with supervised learning was the approach introduced by Zhang et al. [2]. They employed the Long Short Term Memory (LSTM) model and formulated strategies to create optimization objectives from the provided annotations in the TVSum and SumMe datasets. Subsequent works built on top of it, while addressing the challenges posed by LSTMs in terms of modelling long range dependencies. These include approaches with Fully Convolutional Sequence Networks [5], Memory networks [3], Graph networks [13] and the predominant approach, Self Attention [4, 14, 15, 16, 17].

2.2. Temporal Dependencies in Video Data

Several works have reviewed whether deep learning models applied to video data truly learn spatio-temporal relationships. An example of this from Li et al. [9] in activity recognition, demonstrated that due to "representation biases" within the dataset, models can ignore the temporal information within activity recognition benchmark datasets, UCF101 [18] and Kinetics [19] and rely only on static cues to classify activities. Li et al. [8] also demonstrated a means to remove representation biases through dataset resampling. Some works within activity recognition also manipulate the temporal order of frames/shots. Sevilla-Lara et al. [20] highlighted a shuffling approach which aims to identify which videos require temporal information, dubbing them to be "temporal classes". Huang et al. [21] introduced two frameworks to isolate and analyse temporal features within popular activity recognition models.

3. Methodology

Our followed methodology consists of adapting an existing pipeline, using state-of-the-art models, which are trained using a series of temporal perturbation approaches.

3.1. Video Summarization Pipeline

The video summarization pipeline used for this work is adapted from Zhang et al. [2]. In the following subsections we describe the formulatad pre-processing optimization and evaluation approaches.

3.1.1. Pre-processing

Given a video V with a sequence of M frames, we first sub-sample it to a lower frame-rate, typically to 2 frames per second. This step will lead to N, frames, denoted by $N_j, j \in [1, 2, 3, \ldots N]$ frames. Next, these frames are given to a feature extractor F. In this work we are using GoogleNet [22], in order to enable a fair comparison with previous works. This results in a feature representation $F(N_j)$ per frame of the sub-sampled sequence, which are fed to the model for training/inference.

The TVSum and SumMe dataset annotations are pre-processed to create an optimization objective or "ground truth" importance scores. These scores represent what a collection of human annotators deem

to be relevant for a final summary. Both datasets provide multiple human annotations per video. For TVSum, this takes the form of a score between 1 to 5, while for SumMe, it is a score selected between 0 and 1. The ground truth importance score is then computed as the average over all annotators as mentioned by Fajtl et al. [15] in case of the TVSum dataset, while for SumMe, the scores are provided by the dataset creators.

3.1.2. Model

The whole or partial sequence of processed frames are given to the model to predict their "importance scores". This results in a sequence of scores $\hat{y}_j = \text{Summariser}(F(N_j))$ where $\hat{y}_j \in (0, 1)$, while the Summariser is a Deep Neural Network. This work provides the model with the full video as an input in line with previous work with self attention models [23, 4, 15]. The loss function is computed with respect to the ground truth importance scores per frame y_j and depicted in equation 1:

$$\mathscr{L} = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2 \tag{1}$$

3.1.3. Evaluation metrics

The models are evaluated by predicting the importance scores of all frames from a single video. The primary evaluation metrics used in video summarization are the Kendall and Spearman correlation coefficients as proposed by Otani et al. [24]. This choice is due to the fact that the F1 score metric has been demonstrated to be greatly affected by the pre-processing and post-processing pipeline.

Due to the differences in the dataset annotations, the correlation coefficient is computed in two separate ways. In the case of the TVSum dataset, the correlation score of a video is computed as the average correlation over the model predictions with each human scaled score, since the annotations are scores between 1 to 5. In the case of the SumMe dataset, as it provides only a $0/1$ score of whether a shot is included or excluded in a summary, we first compute the average score over all annotators and then we measure the correlation with the model's prediction. For our experiments, we report and compare our results using only the Kendall correlation coefficient.

3.2. Models

3.2.1. Temporally Invariant Models

For this study, the Multi-layer Perceptron (MLP) and VASNet [15] are the chosen baselines. The VASNet model was chosen as it lacks positional encoding, since the authors noted that the positional order may not be relevant for video summarization [15]. However, they provide the model with the whole video as an input, which could already provide a degree of temporal context. Therefore, we utilise this model for testing our hypothesis with respect to temporal modeling. We investigate whether the frame-wise temporal dependencies within a video were important to achieve their success or whether VASNet could rely on spatial features alone to achieve the same performance. The MLPs architecture is described in the supplementary material.

3.2.2. Adapted Models from Literature

We utilise and adapt two models from the literature for our temporal perturbation experiment: VASNet [15] and PGL-SUM [4].

VASNet The VASNet architecture [15] utilises a self attention module alongside a regressor network to predict frame-wise importance scores. Their approach lacks positional encoding which renders the model permutation invariant. For our temporal perturbation experiment, we introduce absolute positional encoding added directly to the processed frame features prior to the self attention module.

PGL-SUM The PGL-SUM model [4] also utilises self attention and positional encoding. This approach uses both local and global attention, fuses the information obtained from both attention branches and incorporates positional encoding directly to the attention matrix. Therefore, this model is an ideal candidate for the temporal perturbation, as by design it should be sensitive to temporal changes.

3.3. Temporal Perturbations

We define two timescales of information as each video is comprised of disjointed shots, "local" information within a shot and "global" which pertains to the order of the shots. Therefore, we study perturbations across both of these defined scales. In this manner we address the question whether video summarization models are affected significantly either by the short term disruptions or by changes to the overall video structure. We formally define five strategies to perturb the order of the video frames for our experiments, described below. We also illustrate the effect different shuffling strategies have on the original sequence in Appendix A.

3.3.1. Shuffling Strategies

Let us consider a dataset composed of NV videos (e.g. $NV = 50$ for TVSum). Each video $V^j, j \in \{1, \dots, NV\}$ is comprised of N^j frames, represented as an ordered set $V^j = \{F_1,\ F_2,\ F_3,\ \dots, F_{N^j}\}$ where $F_i,\ i \in N^j$ is the feature representation corresponding to the video frame with index i.

Flip: First, the ordered set is flipped leading to: $V_f^j = \{F_{N^j},\ F_{N^{j-1}}, F_{N^{j-2}},\ \dots, F_1\}$.

Fixed Segment Shuffles: Next, the ordered set is divided into M fixed segments having a length of N^j/M. For simplicity, lets assume a subset of the original video into a number of $M = 3$ segments with length $M_l = 2$, as $V = \{F_1, F_2\ |F_3, F_4\ |F_5, F_6\}$. A fixed segment shuffle will permute the video as $V_{fs} = \{F_3, F_4\ |F_5, F_6\ |F_1, F_2\}$

Shot Level Shuffles: Since the TVSum and SumMe datasets provide shot boundaries, we also utilise shot level shuffles[1]. Let's assume that given a video V^j of N^s shots, the corresponding representation will be $V^j = \{S_1,\ S_2,\ S_3, \dots, S_{N^s}\}$. Furthermore, each shot is composed of SF frames, such that $S_i = \{F_1, F_2, F_3 \dots F_{SF}\}$, where F_j represents the frame level feature representation at index j. We propose three strategies that manipulate the shot order at various scales while keeping intact the order within the shots:

1. **Intra-shot shuffling:** The overall shot order of the video is retained, but the frames within each shot are shuffled. For simplicity, for a video V of four shots with varying length $V = \{F_1, F_2\ |F_3, F_4, F_5, F_6\ |F_7, F_8, F_9|\ F_{10}, F_{11}\}$, the shuffled video appears as:
   $V_{is} = \{F_2, F_1|\ F_5, F_4, F_3, F_6|\ F_9, F_7, F_8|\ F_{10}, F_{11}\}$.
2. **Neighbouring Shot Shuffling:** Neighbouring shots are shuffled between each other, leading for example to: $V_{ns} = \{S_2, S_1, S_3,\ |S_6, S_4, S_5\ |\ . .\ |S_N, S_{N-1}\}$
3. **Any shot shuffling:** This strategy is similar to the approach proposed within the fixed segment shuffles, while the difference consists of the varying size of a shot. Each shot is shuffled to randomly appear in a different position in the video.

4. Experiments

We conducted two experiments. In the first, we estimated a baseline with temporally invariant models. This allows us a first sense of how much order contributes to the performance on benchmark datasets. In

[1]Since we utilise a sub-sampled input video, but the shot boundaries are defined for videos in the original frame-rate, we decide which frame belongs to which shot based on the frame index and the sampling rate. This choice is further discussed in the supplementary material

the second, we disrupted the temporal order of videos in different ways. This allowed us to investigate the impact of different types of temporal disruptions on performance.

4.1. Experimental protocol

Datasets. The *TVSum* and *SumMe* datasets are used for our experiments since they are the benchmarks in question. We utilised a pre-processed version of each of these datasets as provided by Zhang et al [2]. In this paper, we report the results using the canonical data setting as described by Zhang et al [2]. The description of each dataset is provided in Table 1.

Table 1
Datasets descriptions, according to Apostolidis et al. [11]

Dataset	Duration	Videos	Topics
TVSum	3 - 10	25	news, how-to's, user-generated, documentaries
SumMe	1-6	50	holidays, events, sports

Experimental Design. The experiments were conducted using with the procedure followed in previous studies [15, 4, 25, 26, 14]. Each experiment is conducted using a five-fold cross validation split, where 80% of the videos are used in the training split and 20% of the videos are used in the test split from each of the benchmark datasets. The best correlation scores are recorded for each split and the overall performance is computed as an average over all the splits. We utilise 3 permutations of a five-fold cross-validation split.

Implementation Details. The pre-processing is adopted from Zhang et al. [2], in which each video is sub-sampled to 2 frames per second and each frame undergoes feature extract using GoogleNet [22] to. The full configuration for each of the models and each of the experiments can be found in the supplementary material. All models were trained for 50 epochs, with a weight decay of $1e^{-5}$, using the mean squared error loss function and gradient clipping. The temporally invariant baseline models are trained with a batch size of 128, with a learning rate of $5e^{-5}$, while the temporal perturbation models are trained with a similar learning rate, but with a batch-size of 1.

4.2. Description of experiments

Experiment 1: Temporally invariant Baselines. We first establish a baseline for supervised video summarization relying purely on spatial features by removing any temporal context. Previous approaches [2, 5, 15, 27] typically give a part of the same video, or the whole video to the model for training. In our approach, we sample a batch of frames and ground truth annotations from any video in the training set, wherein the selected frames in the batch can be from different videos and from any time-step. This approach removes any potential temporal context, focusing entirely on the frame content. The model is optimised using the frames' ground truth scores and is evaluated in the same manner as done by previous existing works in video summarization (i.e the model is evaluated by providing the whole test video as an input).

To compare between approaches, we train the MLP and VASNet models described in Section 3.2.1 and we train two temporally aware models, namely, VASNet with positional encoding and PGL-SUM [4]. These models are trained using the original procedure as described in Section 3.1

Experiment 2: Temporal Disruptions. We investigate the effect of temporal disruptions on the performance of a video summarization model. To demonstrate this, we first establish the baseline performance of a model trained on and evaluated with unshuffled data. Then, for each perturbation strategy as proposed in Section 3.3, we train the model on shuffled data and evaluate their performance

on the unshuffled test split. The models used for this study are the PGL-SUM [4] and VASNet [15] with incorporated positional encoding.

5. Results

We conducted two experiments. In the first, we estimated a baseline with temporally invariant models. In the second, we disrupted the temporal order of videos in different ways.

Table 2
Kendall Correlation coefficients from the Temporally Invariance Experiment

Model	TvSum	SumMe
Human Baseline[24]	0.177	-
Temporally Invariant Baseline		
VASNet (-PC)	0.180	0.0545
MLP	0.170	0.065
Temporally Dependent Models		
VASNet (+PC)	0.147	0.082
PGLSum [4]	0.174	0.033

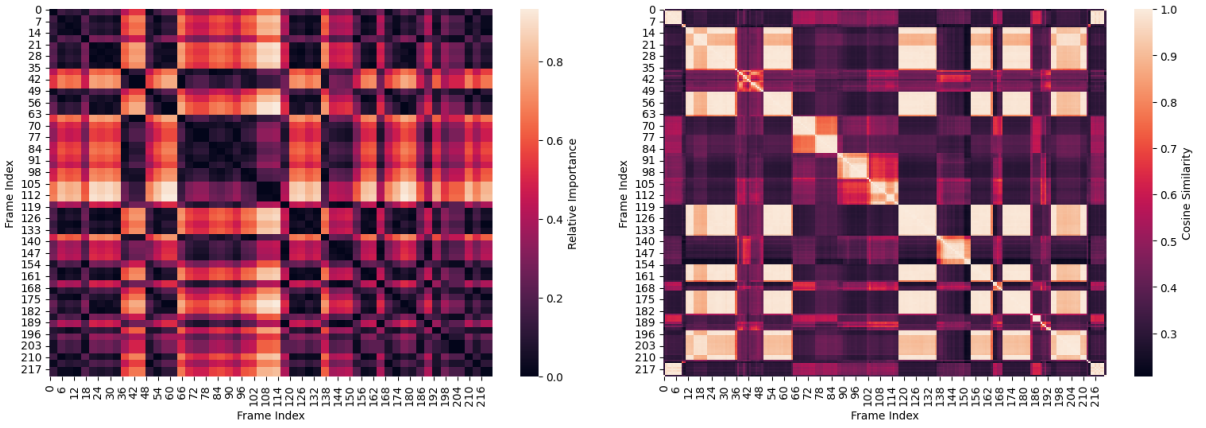
5.1. Temporally Invariant Baseline

The first experiment establishes the Kendall correlation coefficients that the temporally invariant models achieve on the two benchmark datasets for video summarization. We compare their results to existing work within video summarization, the human baselines provided by Otani et al. [24] and with the existing works trained on our procedure to allow comparison. The results, as seen in Tables 2 and 3, show that the MLP baseline is comparable to that of the human baselines (0.170 vs 0.177) on the TVSum dataset. The self-attention model trained with the temporally invariant paradigm described in Section 4.2 also achieved 90% of the performance of the state-of-the-art model MAAM [16] on the TVSum dataset (e.g. 0.180 vs 0.207). It is also worth noting that the Self Attention model trained using our temporally invariant paradigm described in Section 4.2 ($-PC$) outperformed its temporally dependent counterpart ($+PC$) by 19% on the TVSum dataset. This result is notable as the self attention model with the temporally invariant paradigm received frames from different videos. This behaviour can be due to two possibilities, either that the frame level features alone are effective for the TVSum dataset, or that the use of positional encoding directly added to the CNN features may harm the models' prediction capability.

Table 3
Comparison of the Baseline Kendall Correlation Coefficients with other state-of-the-art models

Model	Split type	TvSum	SumMe
Existing Work			
A2Summ[26]	1 × 5 FCV	0.137	0.108
MAAM[16]	1 × 5 FCV	0.179	-
MAAM(VIT)	1 × 5 FCV	0.207	0.227
SSPVS[28]	1 × 5 FCV	0.181	0.192
VHJMT[23]	1 × 5 FCV	0.097	0.106
Clip-It[14]	1 × 5 FRV	0.108	-
SumGraph[29]	1 × 5 FCV	0.094	-
PGL-SUM[4]	1 × 5 FRV	0.150	-
MSVA[27]	1 × 5 FRV	0.190	0.200
Baselines			
MLP	3 × 5 FCV	0.171	0.080
VASNet(-PC)	3 × 5 FCV	0.186	0.067

However, in the case of the SumMe dataset, the results showcase that the VASNet with positional encoding(+PC in the table) narrowly beat VASNet without positional encoding(-PC in the table). This indicates that positional information may play a minor role within this benchmark dataset.



(a) Ground Truth Differences

(b) Framewise Cosine Similarity

Figure 1: A visualization, for one video, of the heatmaps of the pairwise differences between the ground truth scores compared with the frame-wise cosine similarity. Here, frames that are visually similar also appear to have low differences in their importance (note, inverse color).

To further investigate why these temporally invariant models achieve competitive Kendall coefficients, we analyse the processed datasets, in terms of features and labels to check whether there is any relationship between them. Our assumption is that similar features, measured in terms of their representation, will also be assigned similar importance scores. The exact followed procedure consists of creating two heatmaps, one including the pairwise cosine similarity between the features of each frame and the second encompassing the differences between their ground truth scores. We analyzed these relationships using with a few examples taken from the TVSum dataset and in Figure 1, *Video* − 5 is depicted. It can be easily noticed that frames having a high cosine similarity also have small differences in their ground truth importance scores. In contrast, for Video − 32,[2] the rated importance difference did *not* follow the frame-wise similarity.[3] As Video − 5 recorded a higher score than Video − 32 for all of the models, this provides us with a possible explanation for the success of our temporally invariant models – the nature of the datasets.

Table 4
The resulting Kendall Correlation Coefficients(best model in bold) of the temporal disruption experiment, for different models (PGL-SUM, VASNet+PC), and datasets (TVSum, SumMe).

Shuffle	PGLSum		VASNet(+PC)	
	TvSum	SumMe	TVSum	SumMe
Unshuffled	0.174	0.033	0.147	0.088
Fixed Segment	**0.189**	0.085	0.169	**0.138**
Flip	0.176	0.039	0.128	0.883
IntraShot	0.175	0.085	0.175	0.091
Neighbour Shot	0.175	0.062	0.170	0.128
Any Shot	0.190	0.113	0.191	0.125

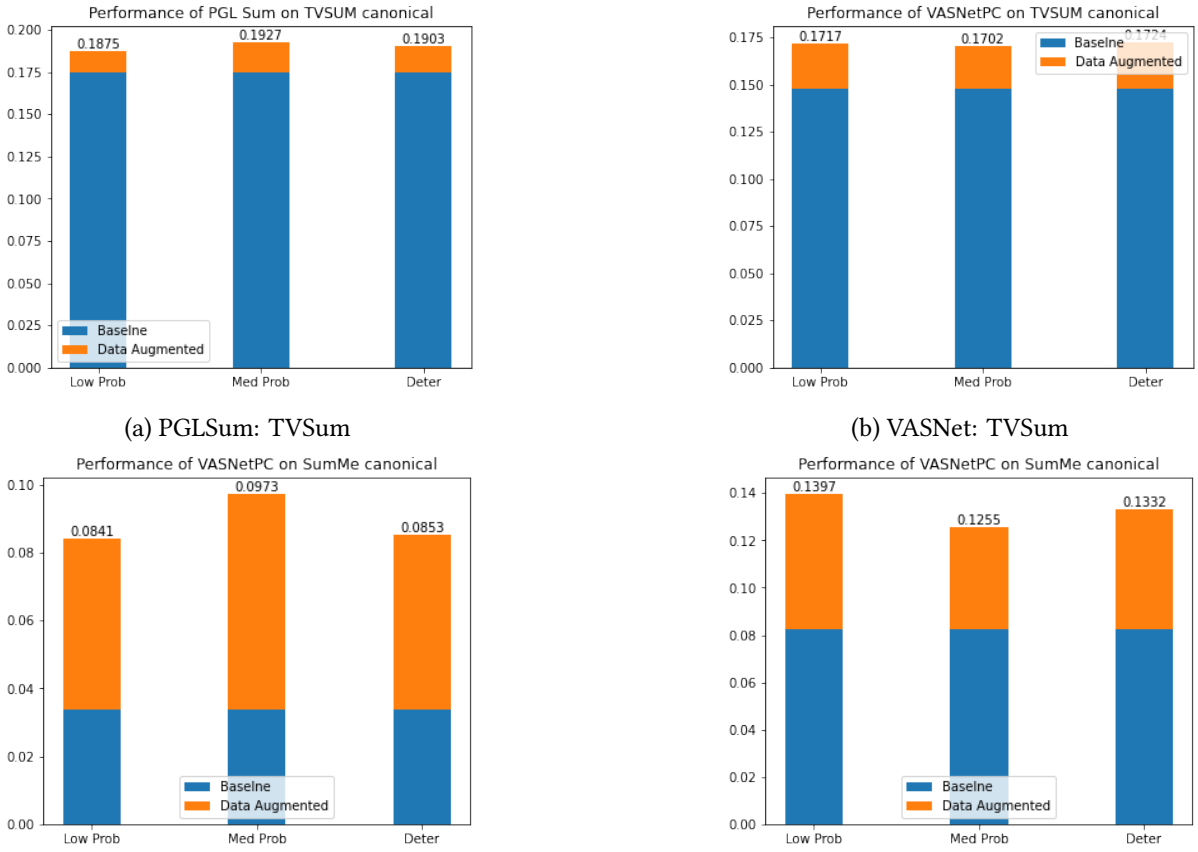
[2]Visualized in the supplementary material due to space limitations.

[3]This may be because Video − 32 depicts a flash mob where each frame possesses similar objects and settings, but the actions and motions within the video are distinct between shots.

5.2. Temporal Disruptions

The second experiment demonstrates the effect that the temporal perturbations described in Section 3.3 have on the performance of temporally dependent video summarization models. The results suggest that some of these strategies show little change over their unshuffled baselines, but show an improvement in strategies that shuffle across fixed time segments. As illustrated in Table 4, the intra-shot, flip, and neighbouring shot shuffles strategies score close to the TVSum baseline performance in both models. In the case of the SumMe dataset, they invariably improve the performance of the model.

The *Fixed Segment Shuffle* and *Global level shot shuffling* improve model performances across TVSum and SumMe. In particular, PGL-SUM shows an improvement on the Kendall correlation coefficient of 9.7% using the *Fixed Segment Shuffle* and 8.6% using the *Global level shot shuffling* on the TVSum dataset. A notable point is that the shuffling strategies improved the correlation score over the SumMe dataset significantly in the case of VASNet with positional encoding, especially when using the Fixed Segment Shuffling, but not as significantly in the case of PGL-SUM.



(a) PGLSum: TVSum

(b) VASNet: TVSum

(c) PGLSum: SumMe

(d) VASNet: SumMe

Figure 2: The Results of the Data Augmentation Experiment. The orange shows the extent to which the models exhibited an improved Correlation Coefficient. As seen here, both PGLSUM and VASNet recorded an improvement, while this improvement was larger in the case of the SumMe dataset.

Temporal Perturbation as Data augmentation. The improvements recorded on the Kendall correlation scores using the *Fixed Segment Shuffles* and *Shot Level Shuffles* strategies could be because they provide different views on the same data. One could assume that permuting fixed shots and segments provides multiple cuts over the same long video, presenting a different narrative each time. In other computer vision tasks such as image classification, such modifications like cutting parts of an image or flipping the image are used as data augmentation strategies to address data limitations. Along these lines, we tested whether these strategies can function as data augmentation since they appear to

work similarly. As an extension to the previous experiment, we combine the flip and the fixed segment perturbations and present the model with a probability of receiving shuffled data. These two strategies were chosen as they introduce a change both globally (*Flip*) and locally (*Fixed Segment Shuffle*). The results in Figure 2 show that both models achieve an improved performance for both datasets.

6. Discussion

Given that the results without temporal information are already comparable to the human baseline (0.170 for the MLP model vs 0.177 for the human baseline in terms of their Kendall correlation in the case of TVSum), we checked for which percentage of the predictions made by one baseline model (in this case the MLP) correlated well with human summaries. Assuming an acceptability threshold for the Kendall correlation informed by the human baseline of 0.15, we see that this is achieved by 52% videos of the TVSum dataset (26 out of 50) and 20% of the SumMe dataset (5 out of 25). This indicates that a notable portion of each dataset may not require temporal context. However, this could also indicate that simply correlating between model predictions and human labels may not adequately measure a model's capacity to summarise videos based on context.

The improvements seen when we introduce the *Shot level Shuffles* and the *Fixed Segment Shuffles* could indicate that short-term temporal context may benefit the model to a certain extent. This behaviour could be explained by the labeling strategy employed by the TVSum dataset, since the annotators were presented with video shots in a random order. The improvements recorded over the SumMe dataset especially highlight the data scarcity issue that plagues supervised video summarization, as a simple strategy improving the performance is quite indicative of this effect.

7. Conclusion

A key aspect of creating video summaries resides in deciding what should be included based on past and future context. Supervised video summarization models should learn to use temporal context to predict what is, and what is not, relevant to the final summary. The results indicate that temporal context provides a limited benefit towards supervised video summarization and that short temporal dependencies may be useful for the TVSum and SumMe benchmark datasets. More crucially, the results of our experiments suggest that models that lack temporal context achieve competitive scores on the video summarization benchmark datasets. Jointly, our findings underscore the need for future work to concretely evaluate the potential static biases that may prevail in these benchmarks. More vitally, we need to consider the temporal bias not only when designing new methods, but also when we create new benchmark datasets. These new benchmark datasets for summarization should also consider temporal information.

Acknowledgments

This publication is part of the project ROBUST: Trustworthy AI-based Systems for Sustainable Growth with project number KICH3.LTP.20.006, which is (partly) financed by the Dutch Research Council (NWO), RTL, and the Dutch Ministry of Economic Affairs and Climate Policy (EZK) under the program LTP KIC 2020-2023.

References

[1] M. Otani, Y. Song, Y. Wang, et al., Video summarization overview, Foundations and Trends® in Computer Graphics and Vision 13 (2022) 284–335.

[2] K. Zhang, W.-L. Chao, F. Sha, K. Grauman, Video summarization with long short-term memory, in: ECCV, Springer, 2016.

[3] J. Wang, W. Wang, Z. Wang, L. Wang, D. Feng, T. Tan, Stacked memory network for video summarization, in: Proceedings of the 27th ACM International Conference on Multimedia, MM '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 836–844. URL: https://doi.org/10.1145/3343031.3350992. doi:10.1145/3343031.3350992.

[4] E. Apostolidis, G. Balaouras, V. Mezaris, I. Patras, Combining global and local attention with positional encoding for video summarization, in: 2021 IEEE International Symposium on Multimedia (ISM), 2021, pp. 226–234.

[5] M. Rochan, L. Ye, Y. Wang, Video summarization using fully convolutional sequence networks, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 347–363.

[6] Y. Song, J. Vallmitjana, A. Stent, A. Jaimes, Tvsum: Summarizing web videos using titles, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5179–5187. doi:10.1109/CVPR.2015.7299154.

[7] M. Gygli, H. Grabner, H. Riemenschneider, L. Van Gool, Creating summaries from user videos, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 505–520.

[8] Y. Li, N. Vasconcelos, Repair: Removing representation bias by dataset resampling, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 9572–9581.

[9] Y. Li, Y. Li, N. Vasconcelos, Resound: Towards action recognition without representation bias, in: European Conference on Computer Vision, 2018. URL: https://api.semanticscholar.org/CorpusID:52255684.

[10] T. T. Um, F. M. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, D. Kulić, Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks, in: Proceedings of the 19th ACM international conference on multimodal interaction, 2017, pp. 216–220.

[11] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, I. Patras, Video summarization using deep neural networks: A survey, Proceedings of the IEEE 109 (2021) 1838–1863. URL: https://api.semanticscholar.org/CorpusID:231627658.

[12] A. Chhabra, K. Patwari, C. Kuntala, Sristi, D. K. Sharma, P. Mohapatra, Towards fair video summarization, Transactions on Machine Learning Research (2023). URL: https://openreview.net/forum?id=Uj6MRfR1P5.

[13] J. Park, J. Lee, I.-J. Kim, K. Sohn, Sumgraph: Video summarization via recursive graph modeling, ArXiv abs/2007.08809 (2020). URL: https://api.semanticscholar.org/CorpusID:220633480.

[14] M. Narasimhan, A. Rohrbach, T. Darrell, Clip-it! language-guided video summarization, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 13988–14000. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/7503cfacd12053d309b6bed5c89de212-Paper.pdf.

[15] J. Fajtl, H. S. Sokeh, V. Argyriou, D. Monekosso, P. Remagnino, Summarizing videos with attention, 2018. arXiv:1812.01969.

[16] H. Terbouche, M. Morel, M. Rodriguez, A. Othmani, Multi-annotation attention model for video summarization, 2023, pp. 3143–3152. doi:10.1109/CVPRW59228.2023.00316.

[17] Y. Jung, D. Cho, S. Woo, I. S. Kweon, Global-and-local relative position embedding for unsupervised video summarization, in: European Conference on Computer Vision, Springer, 2020, pp. 167–183.

[18] K. Soomro, A. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild, ArXiv abs/1212.0402 (2012). URL: https://api.semanticscholar.org/CorpusID:7197134.

[19] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6299–6308.

[20] L. Sevilla-Lara, S. Zha, Z. Yan, V. Goswami, M. Feiszli, L. Torresani, Only time can tell: Discovering temporal data for temporal modeling, in: Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2021, pp. 535–544.

[21] G. Huang, Z. Liu, K. Q. Weinberger, Densely connected convolutional networks, 2017 IEEE

Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 2261–2269. URL: https://api.semanticscholar.org/CorpusID:9433631.
[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
[23] H. Li, Q. Ke, M. Gong, R. Zhang, Video joint modelling based on hierarchical transformer for co-summarization, IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).
[24] Mayu Otani, Yuta Nakahima, Esa Rahtu, and Janne Heikkilä, Rethinking the evaluation of video summaries, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[25] P. Li, Q. Ye, L. Zhang, L. Yuan, X. Xu, L. Shao, Exploring global diverse attention via pairwise temporal relation for video summarization, Pattern Recognition 111 (2021) 107677. URL: https://www.sciencedirect.com/science/article/pii/S0031320320304805. doi:https://doi.org/10.1016/j.patcog.2020.107677.
[26] B. He, J. Wang, J. Qiu, T. Bui, A. Shrivastava, Z. Wang, Align and attend: Multimodal summarization with dual contrastive losses, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.
[27] J. A. Ghauri, S. Hakimov, R. Ewerth, Supervised video summarization via multiple feature sets with parallel attention, in: 2021 IEEE International Conference on Multimedia and Expo (ICME), 2021, pp. 1–6s. doi:10.1109/ICME51207.2021.9428318.
[28] H. Li, Q. Ke, M. Gong, T. Drummond, Progressive video summarization via multimodal self-supervised learning, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 5584–5593.
[29] B. Zhao, H. Li, X. Lu, X. Li, Reconstructive sequence-graph network for video summarization, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (2022) 2793–2801. doi:10.1109/TPAMI.2021.3072117.
[30] V. I. Levenshtein, et al., Binary codes capable of correcting deletions, insertions, and reversals, in: Soviet physics doklady, volume 10, Soviet Union, 1966, pp. 707–710.

A. Differences between Shuffled and Original Video Sequences

We demonstrate the effect of different shuffles through the use of the Levenshtein/edit distance [30]. The Levenshtein distance has been used previously to measure the similarity between two string sequences by quantifying the substitutions, insertions, and deletions to transform one sequence to another. Since our proposed shuffling strategy manipulates the order of a video sequence through the indices, the Levenshtein distance can be used to measure the dissimilarity between them. To demonstrate this aspect, we compute the Levenshtein distance over the entire TVSum and SumMe datasets for different shuffling strategies described in Section 3.3. It is important to note that the Levenshtein similarity is

Table 5
The Levenshtein distance for different shuffling strategies for the TVSum Dataset. A lower score implies that there has been a significant change between the sequences.

Shuffle type	Shuffle Iterations	Levenshtein Distance (scaled to 100)
Flip	1	0.15
Intra Shot Shuffle	3	100.00
Fixed Segment Shuffle	3	11.03
Neighbouring Shot Shuffle	3	58.15
Whole Shot Shuffle	3	6.43

always 0 when comparing a flipped sequence with the original, and will also be 100 when comparing the intrashot shuffled sequence with the original. This is largely due to the way the Levenshtein distance is computed. However, it may not always be the case that the flipped video is semantically dissimilar from

Table 6
The Levenshtein distance of multiple shuffles for the SumMe Dataset. A lower score implies that there has been a significant change between the sequences.

Shuffle type	Shuffle Iterations	Levenshtein Distance (scaled to 100)
Flip	1	0.19
Intra Shot Shuffle	3	100.00
Fixed Segment Shuffle	3	18.79
Neighbouring Shot Shuffle	3	59.71
Whole Shot Shuffle	3	13.22

the original and that videos with their shots shuffled internally have no difference with their unshuffled counterparts.

B. Shot Division in Downsampled inputs

Both the TVSum and SumMe datasets provided shot-boundaries which were created utilising the Kernel temporal segmentation algorithm. These shot boundaries were computed using the full sequence of inputs and given as indices where a shot starts or ends . For e.g Shot 1 could be between $(0, 12)$ can be between the frame indices of 0 to 12. However, the processed sequences used for training and evaluation of the models were sub-sampled by skipping every fifteenth frame. Therefore, to assign which index belongs to which shot, we simply multiply each index by 15 in the sub-sampled sequence and then assigned to which shot boundary in the original sequence based on the boundaries provided. An example of this is illustrated of this as follows; Assume we have a subsampled sequence of 9 frames, lets arrange this as an index $[\ 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8\ ]$, lets assume the shot boundaries provided by the original videos to be $(0, 12), (13, 60), (61, 106), (107, 120)$. Then we multiply the subsampled index by 15 $[0,\ 15,\ 30,\ 45,\ 60,\ 75,\ 90,\ 105,\ 120\ ]$, then the final shots assigned are as $[0,\ 1,\ 1,\ 1,\ 1,\ 2,\ 2,\ 2,\ 2,\ 3\ ]$

C. Experimental Configurations

C.1. Shared Hyperparameters

All of the models were trained with the following set of hyperparameters shared between them

1. Epochs: 50
2. Weight decay: 1e-5
3. Gradient Norm Clipping: 3
4. Learning rate: 5e-5
5. Optimizer: Adam
6. Input Dimensions: 1024

C.2. MLP and Attention

For the temporally invariant baselines, the hyper-parameters for each model are listed as follows. The main observations relate to the batch size of 128 with a learning rate of 5e-5 and the ADAM optimizer to train the model.

- Self Attention configuration
  - Self Attention Linear projection dimension: 1024
  - FeedForward Neural Network dimensions: 1024
  - Number of heads: 1
  - Drop-out: 0.5

The Multi-layer perceptron architecture used in this work is described as follows:

```
class MLPM(nn.Module):
    def __init__(self,dim=1024,pos_enc=False):
        super(MLPM,self).__init__()
        self.m = 1024
        self.hidden_size = dim
        self.ka = nn.Linear(in_features=self.m, out_features=1024)
        self.kd = nn.Linear(in_features=self.ka.out_features, out_features=1)
        self.layer_norm_ka = LayerNorm(self.ka.out_features)
        self.relu = nn.ReLU()
        self.drop50 = nn.Dropout(0.5)
        self.sig = nn.Sigmoid()
        self.pos_enc = pos_enc
```

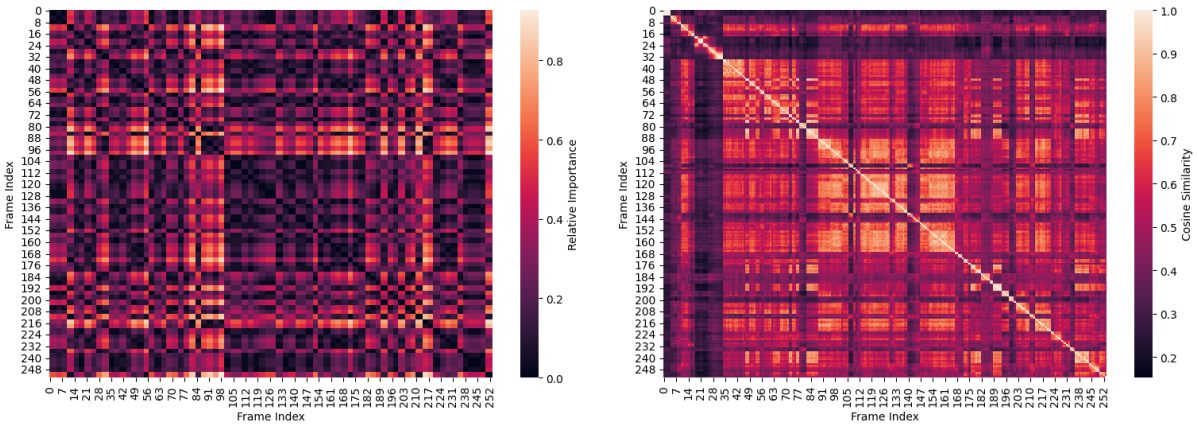
C.3. PGL-SUM

The PGL-SUM architecture was directly taken from the code provided by Apostolidis et al[4]. We take the configuration specified in the code which is as follows

- Number of heads:
  - Local attention: 4
  - Global attention: 8
- Number of Segments: 4
- Absolute positional encoding frequency : 10000
- Fusion Strategy: Addition
- Drop-out: 0.5

D. Cosine Similarity versus Ground Truth Differences visualizations

These are some of the visualizations of the frame-wise cosine similarity of the CNN features of a video versus the absolute differences between the ground truth importance score given in each of the datasets. We chose these examples to showcase the relationships between them



(a) Ground Truth Differences

(b) Framewise Cosine Similarity

Figure 3: Heatmap of Video 32 of the TVSum Dataset