

# Appendix: Supplementary Analysis and Experiments

Guande Wu  
guandewu@nyu.edu  
Jianzhe Lin  
jianzhelin@nyu.edu  
Claudio T. Silva  
csilva@nyu.edu

New York University  
New York, USA

## Appendix A: Implementation

In this appendix, we will describe our implementation of the model and training procedure. Our code is based on Pytorch[1].

### Model

#### Summarizer

The summarizer has the same effects as the selector LSTM in [2], while our summarizer incorporates three sources of features *i.e.* entity-relation aware features extracted by STGCN, the visual scene features processed by an LSTM and difference attentions as we describe in Section 3.1. Below, we will describe the implementations of each sub-module.

**STGCN** Our STGCN module relies on an object detection module to extract the objects and their features. We implement the object detection module upon Facebook Detectron2 [3]. Specifically, we apply a Fast R-CNN with ResNet-50 pre-trained on Microsoft COCO dataset[4]. After assembling the extracted objects into the Spatio-Temporal Graph, we apply a three-layer Graph Convolution Network on the graph. Each layer has a hidden size of 256 and a shortcut connection from the precursory layer. After processing the graph, we perform average pooling on the vertexes of each frame.

**Difference Attention** Our difference attention module is identical to the original one in [2].

**Visual Scene Feature Extraction** We employ GoogLeNet to extract the visual scene features from the videos. We explot the preprocessed feature file provided by [5]<sup>1</sup>. Then we process the features by an LSTM identical to the selector LSTM of [2].

**MLP for Feature Fusion** We employ a Multi-Layer Perceptron (MLP) to fusion the features. Our MLP has three layers, each of which has 128 hidden units.

## Encoder and Decoder LSTMs

Our encoder LSTM, decoder LSTM are identical to the original versions in [9].

## Critic

Our critic consists of two modules *i.e.* LSTM and Video Patch Module described in Section 3.3. The LSTM is identical to the discriminator LSTM in [9]. As we mentioned in Section 3.3.2, Video Patch Module comprises of  $M$  building blocks. In practice, we set  $M = 3$ .

## Losses

In the training process, we employ a series of loss functions. Below, we present their definitions.

**Score-Sum Loss** Our loss function for Score-Sum Loss is defined as:

$$L_{sum} = \frac{\sum s_t}{\sqrt{T}} \quad (1)$$

**Sparsity Loss** We also employ the sparsity loss defined in [9] as:

$$L_{sparsity} = \left\| \frac{1}{M} \sum_{t=1}^T s_t - \sigma \right\| \quad (2)$$

where  $\sigma = 0.15$  which is same with [9].

**GAN Loss** The GAN loss for the critic is defined as:

$$L_{GAN} = \|E(c(\mathbf{x}')) - E(c(\mathbf{x}))\|_2 + \lambda (\|\nabla c\|_2 - 1)^2 \quad (3)$$

where  $\mathbf{x}$  is the original features and  $\mathbf{x}'$  is the reconstructed features.  $c$  represents the function of the critic.

**Reconstruction Loss** Our reconstruction loss is identical to the vanilla VAE as:

$$L_{reconst} = \frac{\|\mathbf{x}' - \mathbf{x}\|_2}{2} \quad (4)$$

where  $\mathbf{x}$  is also the original features and  $\mathbf{x}'$  is the reconstructed features.

**Prior Loss** Our prior loss  $L_{prior}$  is also identical to the vanilla VAE and [9].

## Training Procedure

In this section, we specify the learning of parameters of 1. Summarizer  $\theta_s$  2. Encoder LSTM  $\theta_e$  3. Decoder LSTM  $\theta_d$  4. Critic  $\theta_c$  Following [9] we train the models via three steps.

1. Optimizing  $\{L_{reconst} + L_{prior} + L_{sparsity} + L_{sum}\}$  to update the parameters  $\theta_s$  and  $\theta_e$ .
2. Optimizing  $\{L_{reconstruct} + L_{GAN}\}$  to update the parameters  $\theta_d$ .
3. Optimizing  $-L_{GAN}$  to update the parameters of  $\theta_c$ .

We fulfill the training procedure on an NVIDIA RTX-8000. It takes us approximately 30 hours to train the models (of 5 splits) on TVSum and 16 hours on SumMe. We train our models with Adam optimizers with a learning rate of  $1e-4$  and 0.1 times after ten epochs.

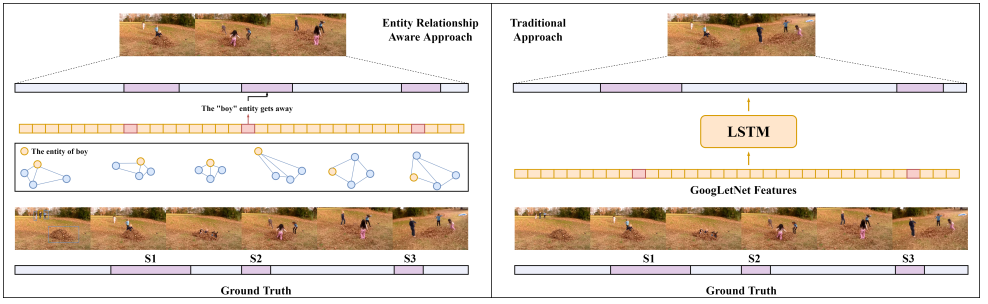


Figure 1: The clip of first 20 seconds of Kid’s playing video in SumMe. The ground-truth summary has three shots in the period, and we denote them as  $S1, S2, S3$ . When the traditional method (SUM-GAN) can capture  $S1$  and  $S3$ , it falls short in capturing  $S2$ .  $S2$  is salient because it records the boy’s running away from the leaf stack after he got "attacked" by the two girls. The boy’s movement can only incur minor visual changes from the perspective of the whole scene. Thus the traditional method relying on the scene-level features may fail to capture it. By comparison, the Spatio-Temporal Graph can extract the object-level features and represent the movement by changing the boy and the other two girls’ entity relationship. In the Figure, Our ERA approach successfully captures  $S2$  when the traditional method not.

## Appendix B: Case Study

In this appendix, we demonstrate the importance of entity relationship by two cases.

### Case 1: Kids’ Interaction

We first further describe the case shown in our introduction. The video depicts a scene where three kids are playing around a leaf stack. We clip the first 20 seconds of the video and demonstrate the advantages of our proposed ERA method based on it. As Figure 1 shows, the clip comprises three key shots according to the ground truth frame scores *i.e.*  $S1, S2$  and  $S3$ .  $S1$  shows that three kids run towards the leaf stack and start playing with each other. Then,  $S2$  records the boy’s running away from the leaf stack because he got "attacked" by the two girls. Finally,  $S3$  describes that the boy runs back to and hits back the two girls. The summary generated by the ground-truth scores includes all three shots. Though the traditional method can also capture  $S1$  and  $S3$ , we observe that it can fall short in capturing  $S2$ .  $S2$  shares similar backgrounds and entities with previous shots as they all have three kids playing around a leaf stack. However,  $S2$  is salient because of the boy’s movement away from the two girls. Such movement only induces minor visual changes from the perspective of the whole scene. Thus, the traditional method relying on the scene-level features can find it challenging to capture the trivial visual change. By comparison, the movement can change the spatial relationship between the boy and the two girls in Spatio-Temporal Graph. Thus, our method can extract the moving event by the object-level features and successfully include  $S2$  in the summary. In this case, the relationship between the entities boy and the other two girls plays a profound role in recognizing the importance of  $S2$ . By contrast, the absence of the entity-relationship may lead to the failure of the clip summarization.



Figure 2: TVSum Video-11: We compare the summaries generated by SUM-GAN (a), ERA (b) and ground-truth frame scores (c).  $B$  is a key shot, describing the staff lifts a platform and pets the dog on the platform. Our method and ground-truth summary capture the shot while SUM-GAN not. Figure (d) shows the frames in the shot  $B$ .

## Case 2: Taking Care of A Dog

Then, we study an essential shot in TVSum Video-11, which also reflects the necessity of using the entity-relationship aware method. The video introduces a pet store and describes how the staff takes care of a dog. The shot  $B$  refers to a scenario when the staff lifts a platform and pets the dog on the platform, as Figure 2 (d) shows.  $B$  is an essential shot because the action is a precursory step for the latter dog-cleaning work. The summary generated by the ground-truth scores also includes the shot shown in Figure 2 (c). However, the baseline SUM-GAN fails to capture the shot shown in Figure 2 (a). The cause can be that the action of lifting the platform and petting the dog is minor from the perspective of the entire scene so that the scene-level visual features extracted by GoogLeNet can not capture the action. By comparison, in our ERA method, the temporal relationship can capture the movement of the platform, and the spatial relationship can capture the petting action. Thus, our method will assign a high score to the shot and include it in the summary, as Figure 2 (b) shows.

## Appendix C: Error Analysis

In this appendix, we analyze the errors of our method. We identify the following common sources of error by comparing our summarized videos on TVSum and SumMe with the ground truth.

**Limited Object Detection Accuracy** Since our method relies on the object detection algorithm to build the Spatio-Temporal Graph, the algorithm’s accuracy strongly affects our method’s performance. When object detection fails for various reasons, e.g., low resolution, our method can also fail to capture the essential shots. We find our method fails due to the limited object detection accuracy in SumMe 15, TVSum-15 and TVSum-21.

**Video Titles** Video titles are common at the beginning of some videos, especially in TVSum. These titles may be a YouTuber’s logo and are not necessarily associated with the video content. However, the titles often deviate from the video content by the visual features. Thus, our method and other state-of-the-art models can get confused by them and fail to exclude them in the summary. We observe this error in TVSum-5, TVSum-15, TVSum-32, and TVSum-48.

**Captions in the Videos** The text caption accompanies some videos. We observe that our method may fail when encountering such video scenes. An example is TVSum-5, in which a

caption is used to conclude the video content while the visual content is akin to the previous shots. Our method fails to capture the shot.

**Abrupt visual changes** The video quality is strongly affected by the shooting conditions. Camera movement and focus can lead to abrupt visual changes in the videos. Those abrupt visual changes can cheat the model to think this is an important segment when it does not mean anything. Such a phenomenon can be observed in SumMe-32 and TVSum-8.

## Appendix D: Training Stability Analysis

In this appendix, we compare the training stability of our method and the vanilla GAN. As Section 4.3.2 shows, using W-GAN and Patch mechanism only brings minor improvement on the baseline SUM-GAN compared to our ERA method. However, W-GAN can still make the training process steadier. To verify it, we report the training process of the different discriminator-side methods in Figure 3.

We choose to report the reconstruction loss in the test dataset since it reflects the distance between the original and reconstructed visual features. We avoid using GAN loss because its definition varies in W-GAN and vanilla GAN. From Figure 3, we observe there is a jump of reconstruction loss after 400 steps in all five splits of the vanilla GAN. By comparison, the jump is not observed in both W-GAN and W-GAN with patch mechanism. Thus, we confirm that our method can be advantageous for the steadier training process.

## Appendix E: Analysis of Video Patch

We introduced the video patch mechanism to cope with varying video lengths in Section 3.3.2. Though the mechanism is proved to be beneficial for the overall performance in our ablation study (Section 4.3.1), its effects on the long videos are not confirmed. To address it, we further analyze the mechanism’s effects on the videos of different lengths. We compare two variants of the STGCN model, which are trained by W-GAN and W-GAN with a patch mechanism (W-GAN-Patch). Since the only difference is the availability of patch mechanisms, the comparison can reflect the effects of the mechanism. Firstly, we rank the SumMe videos by their lengths and derive the F-measure values of the two variants on each video. Then, we calculate the distance of the values and visualize them in Figure 4. From Figure 4, we find that the longer videos (right of X-axis) tend to have positive values while the short videos can be both negative and positive. Since the Y-axis value reflects the performance improvement of W-GAN-Patch over W-GAN, the observation shows that W-GAN-Patch can improve the performance on the longer videos, which confirms our theoretical analysis in Section 3.3.2.

## References

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [2] Yunjae Jung, Donghyeon Cho, Dahun Kim, Sanghyun Woo, and In So Kweon. Discrim-

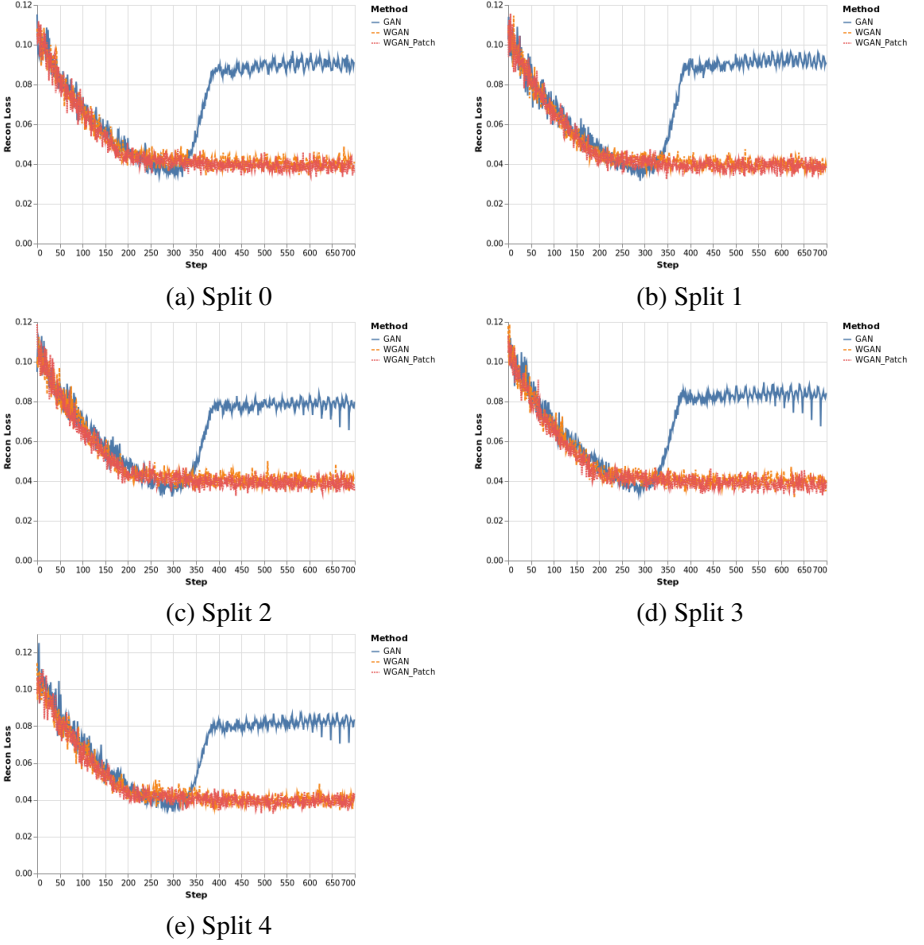


Figure 3: Comparison of the reconstruction loss in the training process of different discriminator-side methods. The experiments are conducted on the five non-overlapping splits of SumMe. GAN refers to the vanilla GAN; WGAN\_PATCH refers to W-GAN with patch mechanism. From the Figure, we observe that there is a jump of reconstruction loss after 400 steps in all five splits of the vanilla GAN. By comparison, the jump is not observed in both W-GAN and W-GAN with patch mechanism.

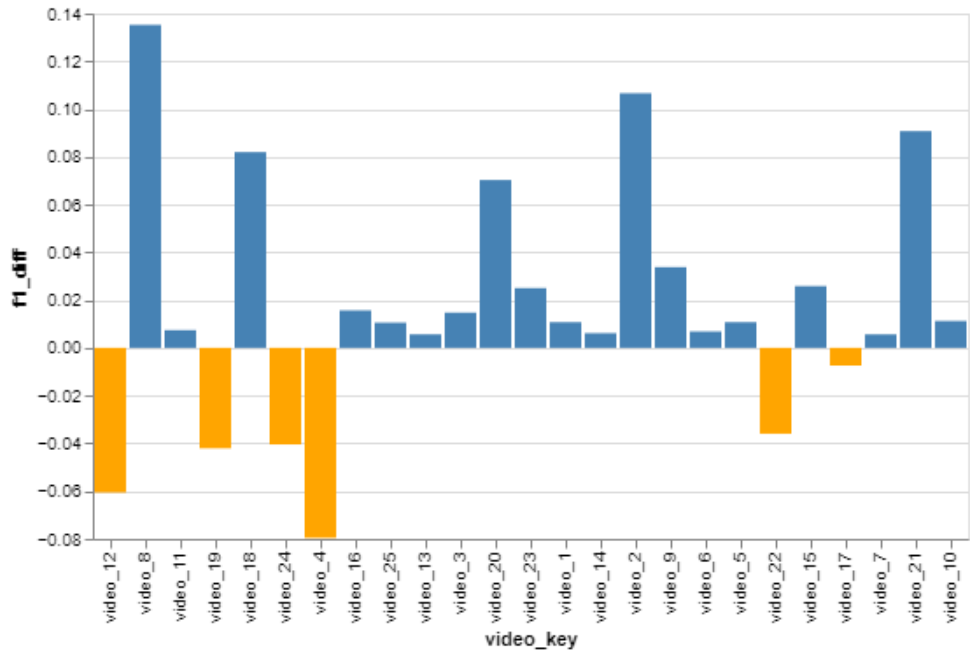


Figure 4: Performance improvement of video patch mechanism. The Y-axis value corresponds to the subtraction of W-GAN F-measure from W-GAN-Patch F-measure. The videos on X-axis is arranged by the increasing order of the video length. The righter the video is arranged, the longer the video is. The figure shows that the W-GAN-Patch tends to improve the performance on the right side videos, which have a longer length.

- inative feature learning for unsupervised video summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8537–8544, 2019.
- [3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.
- [4] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 202–211, 2017.
- [5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.
- [6] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.