The Supplementary of Towards Global Video Scene Segmentation with Context-Aware Transformer

Yang Yang[1,2,3] **Yurui Huang**[1], **Weili Guo**[1*], **Baohua Xu**[4], **Dingyin Xia**[4]

[1]Nanjing University of Science and Technology, [2]MIIT Key Lab. of Pattern Analysis and Machine Intelligence, NUAA,[3]State Key Lab. for Novel Software Technology, NJU,[4]HUAWEI CBG Edu AI Lab

Algorithm Details

In this section, we describe the details of the proposed self-supervised learning scheme by applying pseudo-boundary in Algorithm 1 and Algorithm 2.

Algorithm 1: SSL for Video Scene Segmentation.

```
Input: Shot encoder $f_e$, CAT $f_t$, videos $\mathbf{v}$, Hyperparameters: $L, P, N, \mu$, number of epoch $M$
for $i \leftarrow 1$ to $M$ do
    Sample mini-batch videos;
    Obtain shot representations $\bar{\mathbf{v}} \leftarrow f_e(\mathbf{v})$;
    Generate pseduo-boundaries by Algorithm 2;
    Obtain contextual representations $\hat{\mathbf{v}} \leftarrow f_t(\bar{\mathbf{v}})$;
    Three non-overlapping pseudo-scene sequences $Q_t^{left} = \{\hat{\mathbf{v}}_{t-(P-1)/2}, \cdots, \hat{\mathbf{v}}_{start-1}\}, Q_t = \{\hat{\mathbf{v}}_{start}, \cdots, \hat{\mathbf{v}}_{end}\}$ and $Q_t^{right} = \{\hat{\mathbf{v}}_{end+1}, \cdots, \hat{\mathbf{v}}_{t+(P-1)/2}\}$.
    Calculate loss $L_{smm}, L_{som}, L_{gsm}$ and $L_{lsm}$ by Eq. (4) - (7);
    Calculate pre-training loss $L = L_{smm} + L_{som} + L_{gsm} + L_{lsm}$;
    Update encoder $f_e$ and $f_t$ by gradient descent;
end for
Output: $f_e$ and $f_t$.
```

Algorithm 2: Similarity-based pseudo-boundary generation.

Input:

- Input video $\mathbf{v}$, Hyperparameters: P, μ

```
for $t \leftarrow 1$ to $T$ do
    Construct shot sequence $\mathbf{c}_t = \{\bar{\mathbf{v}}_{t-(P-1)/2}, \cdots, \bar{\mathbf{v}}_t, \cdots, \bar{\mathbf{v}}_{t+(P-1)/2}\}$;
    $sim \leftarrow []$;
    for $i \leftarrow t-(P-1)/2$ to $t+(P-1)/2$ do
        $sim \leftarrow \{sim; cos\,(\bar{\mathbf{v}}_i, \bar{\mathbf{v}}_t)\}$;
    end for
    $start \leftarrow t, end \leftarrow t$;
    for $i \leftarrow t-1$ to $t-(P-1)/2$ do
        if $sim\,[i] > \mu$ then
            $start \leftarrow i$;
        else
            break;
        end if
    end for
    for $i \leftarrow t+1$ to $t+(P-1)/2$ do
        if $sim\,[i] > \mu$ then
            $end \leftarrow i$;
        else
            break;
        end if
    end for
end for
```

Experimental Analysis

Illustration of Shot Sampling

In detail, a shot contains continuous frames taken by the camera without interruption, and a scene is composed of successive shots and describes a same short story. Considering that frame data is highly redundant because there are many repetitive frames, we concentrate on the shot-level segmentation following (Rao et al. 2020; Chen et al. 2021; Wu et al. 2022; Mun et al. 2022). To construct the shots, we follow the sampling strategy in (Rao et al. 2020; Chen et al. 2021). Specifically, the video is sliced according to the shot boundaries, which are determined by the transition of visual modality. Based on the beginning and ending positions of shots, a fixed number of frames (i.e., 3 frames) are selected as the original feature for one shot, i.e., starting, middle, and ending frames.

Additional Implementation Details

Datasets. Traditional OVSD (Rotman, Porat, and Ashour 2017) (i.e., including 21 videos) and BBC planet earth (Baraldi, Grana, and Cucchiara 2015) (i.e., including 11 videos) are small in size and lack rich story lines, lacking real values (Rao et al. 2020). Therefore, MovieNet (Rao et al. 2020) dataset published 1,100 movies where 318 of which are annotated with scene boundaries, another dataset AdCuepoints (Chen et al. 2021) includes 3,975 movies and TV episodes, 2.2 million shots, and 19,119 manual labels. Note

[*]These authors contributed equally.

that only MovieNet released the dataset, so we experiment with this dataset.

Actually, there are two annotation versions of video scene segmentation task for MovieNet: 1) MovieScenes (Rao et al. 2020; Chen et al. 2021) has only 150 annotations which is used in earlier methods, but it is no longer available. 2) MovieScenes-318 is with a total of 318 annotations, i.e., MovieNet. It is notable that movies in the MovieScene (Rao et al. 2020) are all included in MovieNet (Huang et al. 2020).

Table 1: Ablations of the threshold μ in predicting pseudo boundary. The best scores are in bold.

Threshold μ	AP	mIOU	AUC	F1
$\mu = 0.0$	58.67	51.42	91.38	50.54
$\mu = 0.1$	59.07	52.37	91.42	51.57
$\mu = 0.2$	59.23	53.02	91.72	51.42
$\mu = 0.3$	**59.55**	**53.67**	**91.81**	**51.94**
$\mu = 0.4$	59.45	53.62	91.77	51.94
$\mu = 0.5$	59.21	53.42	91.54	51.55

Table 2: Ablations of the pseudo-boundary prediction. The best scores are in bold.

Location	AP	mIOU	AUC	F1
After CAT	59.15	52.95	91.62	51.40
Before CAT	**59.55**	**53.67**	**91.81**	**51.93**

Table 3: Performance comparison with respect to the number of pre-training epochs. The best scores are in bold.

Epochs	2	4	6	8	10
AP	57.54	58.75	59.26	**59.64**	59.59

Introduction of Baseline Approaches

The comparison models fall into three categories: 1) unsupervised methods, i.e., GraphCut (Rasheed and Shah 2005), SCSA (Chasanis, Likas, and Galatsanos 2009), DP (Han and Wu 2011), StoryGraph (Tapaswi, Bäuml, and Stiefelhagen 2014), Grouping (Rotman, Porat, and Ashour 2017). 2) supervised methods, i.e., including Siamese (Baraldi, Grana, and Cucchiara 2015), MS-LSTM (Huang et al. 2020) and LGSS (Rao et al. 2020), and 3) self-supervised methods, including ShotCoL (Chen et al. 2021), SCRL (Wu et al. 2022), and BaSSL (Mun et al. 2022):

- **GraphCut** constructed a weighted undirected shot similarity graph for clustering shots into scenes;
- **SCSA** clustered shots into groups based only on their visual similarity;
- **DP** proposed the multiple normalized min-max cut scores which consider not only neighboring but also non-neighboring scene similarities;

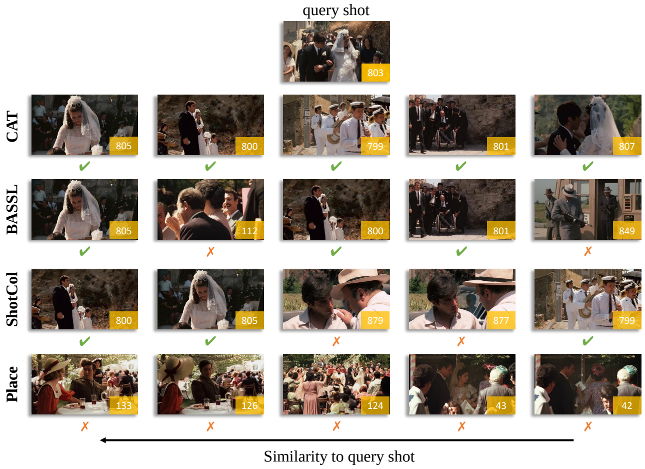

Figure 1: The visualization results of shot retrieval with 5 nearest neighbor shots for a query shot. Shot indices are at bottom-right. Green tick marks indicate shots from the same scene, while yellow cross marks denote shots from a different scene.

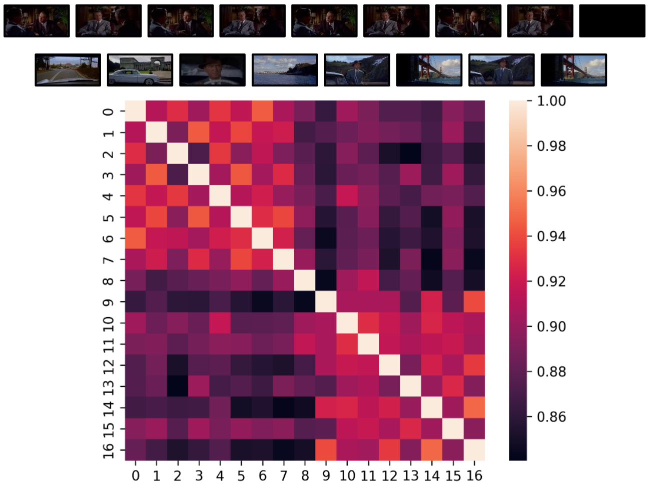

Figure 2: (Best view in color.) Visualization of similarity between shot representations within consecutive shots. We can clearly observe that the local-global encoders can learn effectively contextual information.

- **StoryGraph** conducted scene segmentation by visualizing character interactions as a chart;
- **Grouping** formulated the video scene segmentation as a generic optimization problem to optimally group shots into scenes;
- **Siamese** divided shots by learning a distance measure between shots;
- **MS-LSTM** proposed a Bi-LSTM based model for video scene segmentation;
- **LGSS** integrated multi-modal information across three levels for supervised training;
- **ShotCoL** presented a self-supervised shot contrastive

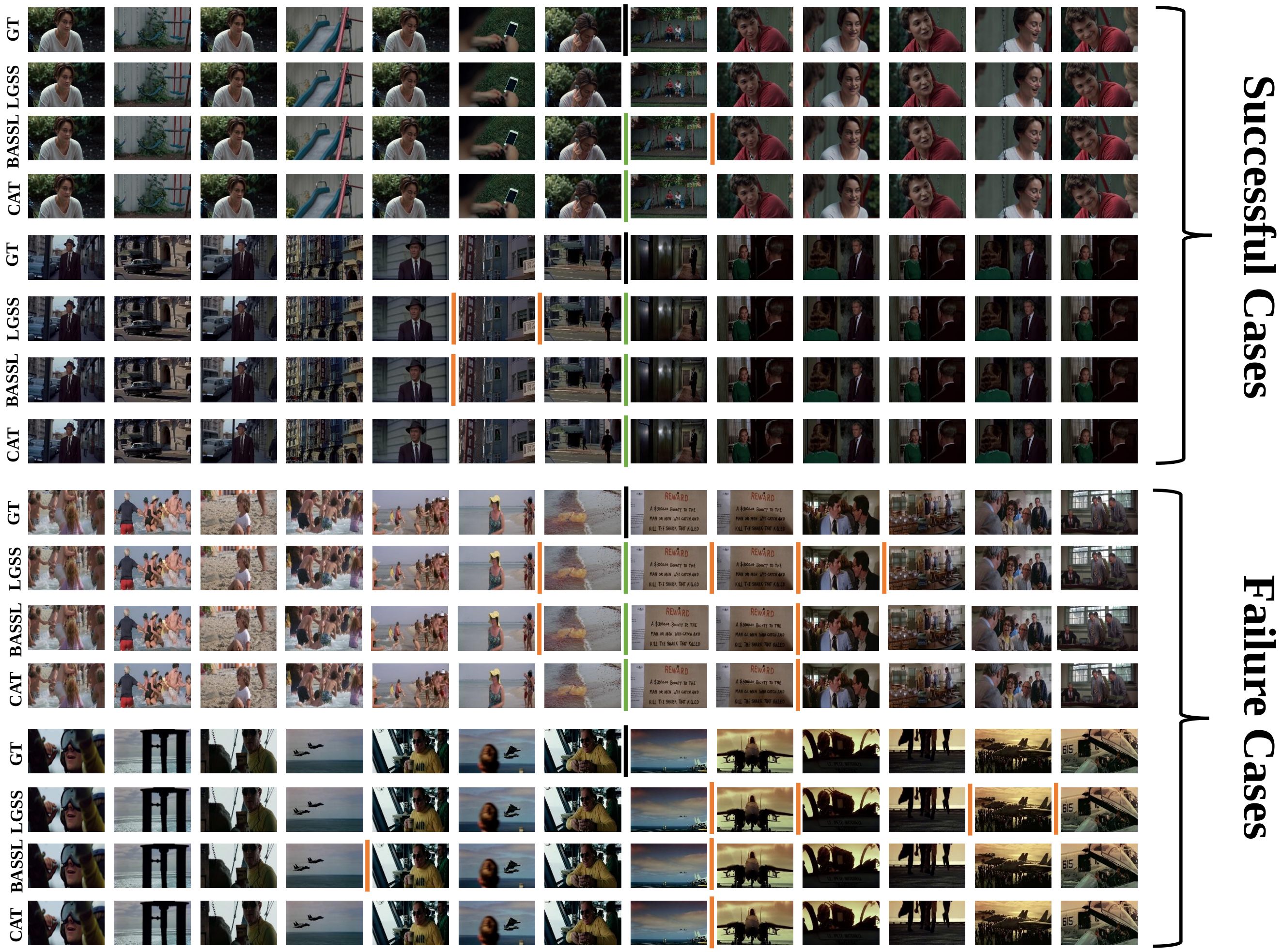

Figure 3: Comparison of pseudo-boundary predictions. GT denotes the ground-truths, green dividing lines indicate the correct boundaries, while the yellow lines denote incorrect ones.

learning approach to learn a shot representation that maximizes the similarity between nearby shots compared to randomly selected shots;

- **SCRL** presented a self-supervised learning scheme to achieve scene consistency, while exploring considerable data augmentation and shuffling methods to boost the model generalizability;
- **BaSSL** tackled video scene segmentation with a self-supervised learning framework in which effective pretext tasks are designed;

Comparison With Existing Self-Supervised Learning Methods

As discussed in the main body, our approach designs a novel context-aware Transformer with local-global self-attention, and uses the self-supervised scheme for training. Therefore, our approach is distinguishable from the shot-level pre-training methods (Chen et al. 2021; Wu et al. 2022), which concerns the shot representation learning by taking a pair of two shots as an input, resulting in a shot encoder like f_e. The closest method to our approach is the BaSSL (Mun et al. 2022), which designs the f_t by directly adopting the Transformer, and utilizes the self-supervised scheme. Firstly, the difference between CAT and BaSSL is the f_t that we design an applicable Transformer for video data. The results in Table 1 of main body also validate the effectiveness, i.e., the CAT performs better than the "CAT with Transformer". Moreover, our pretext tasks are different from BaSSL, which aims to capture the shot-to-scene level information. BaSSL employs the existing DTW technique to acquire the pseudo-boundaries for the output of shot encoder, whereas we design the similarity measure. The results in Table 1 of main body validate the effectiveness, i.e., "CAT with Transformer" performs better than the BaSSL.

Additional Implementation Details

First, we gave the details of shot encoder (i.e., ResNet50) and context-aware Transformer (i.e., CAT). Shot encoder inputs 3 frames for each shot and outputs the shot representations by averaging representations of the 3 frames. For CAT, the hyper-

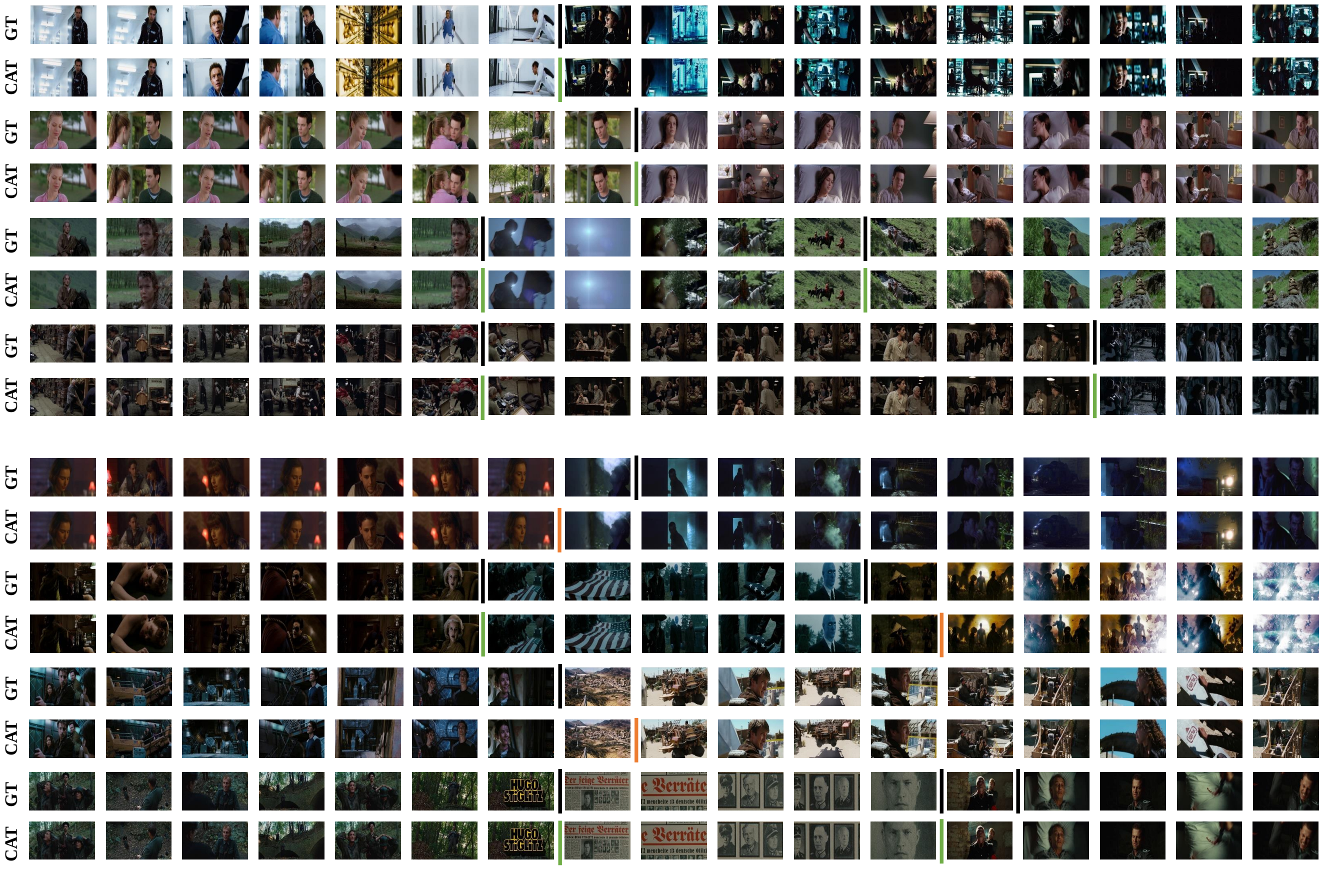

Figure 4: Comparison of boundary detection results from three approaches: LGSS, BaSSL, and CAT. The green dividing lines indicate the correct boundary, while the yellow lines denote incorrect ones. GT denotes the ground truth boundary.

parameters are set to $(H = 2, d_1 = 2048, d = 768, N = 8)$ where H, d_1, d, and N denote the number of layers, the dimension of shot encoder output, the dimension of hidden activation and the number of attention heads, respectively. We apply the Dropout technique (Hendrycks and Gimpel 2016) on hidden states and attention weights with a probability of 10%.

Moreover, we provide the details of pre-training and fine-tuning. The model parameters are randomly initialized for the pre-training process, and then trained using the proposed pretext tasks. We train the model with a batch size of 56 shot sequences, a base learning rate of 0.065, momentum of 0.9, weight decay of 10^{-6}. We pre-train the model for 10 epochs with a linear warm-up strategy for 1 epoch followed by learning rate decaying with a cosine schedule. For the fine-tuning process, we initialize the parameters of shot encoder and the CAT with the pre-trained models. We freeze the shot encoder following (Mun et al. 2022; Chen et al. 2021). We fine-tune the CAT and boundary detection head for 20 epochs using Adam optimizer (Kingma and Ba 2015) with a learning rate of 10^{-5}, and set the batch size as 1024. The learning rate is decayed with a cosine schedule.

Influence of Parameter μ

Table 1 shows the results using different threshold values. The performance of all criteria promotes with the increase of μ and achieves the best performance when μ sets as 0.3, then decreases after 0.3, for the reason that larger μ may add the risk of false pseudo boundaries.

Influence of Pseudo-Boundary Prediction

We conduct more experiments to verify the problem that where is suitable for predicting pseudo-boundary. In detail, “After CAT” denotes predicting pseudo-boundaries with output representations of CAT encoder, and “Before CAT” represents predicting pseudo-boundaries with output representations of shot encoder (i.e., our setting). Table 2 records the results, in which “Before CAT” performs better than “After CAT”. This phenomenon reveals that adopting the output representations of CAT to calculate the similarity will create more noise, considering the integration of contextual information in the forward process.

Influence of Pre-training Epochs

To explore the influence of pre-training epochs, we conduct more experiments. Table 3 exhibits the performance with respect to the number of pre-training epochs. The results find that performance increases until certain numbers and decreases afterward, for the reason that model may over-fit to the noisy pseudo-boundaries.

Visualization Analysis

In this section, we exhibit more cases to validate the effectiveness and limitation of the proposed CAT.

Shot Representations. Figure 1 provides a qualitative example in which 5 nearest neighbor shots are given for a query shot using various shot representations. Almost none of the retrieved results are from the query shot's scene using LGSS features (i.e., Place), even though they are visually similar to the query shot. In contrast, results from CAT representations are all from the same scene even though the appearances do not precisely match, and are more precise than other self-supervised methods. This indicates that CAT can encode local-global context effectively.

Visualization of Local-Global Encoders To validate the effect of local and global encoders, we visualize the matrix of similarity between consecutive shots within a $P-$size window, the results are recorded in Figure 2. We find that the local clusters are clearly obtained, and the similarities of nearby shots are higher than that of distant shots. This validates the effectiveness of the local encoder. Besides, several long-term shots also have high similarity with the anchor, which reveals that the global encoder can provide complementary information to promote the shot learning. Moreover, the 7-th shot is a boundary, so it has high similarities with its left neighbors (i.e., intra-scene) and low similarities with its right neighbors (i.e., inter-scene), which validates the discriminant of the learned shot representations. The 8-th shot is a black transition, thereby it is not similar to its neighbors with low similarity.

Pseudo-boundaries. To validate the quality of discovered pseudo-boundaries with the ground-truths, we record the predictions and ground-truth scene boundaries in Figure 3. In most cases, we observe the pseudo-boundaries identified by the similarity measure are successfully predicted. This phenomenon validates that the pseudo-boundary as a temporal label is useful for self-training. Meanwhile, we also illustrate several failure cases. We find that although predicted pseudo-boundaries do not match the ground truths, they are located close to the ground-truths. Besides, the mismatch may be caused by the noise semantics in the ground truth (e.g., the first case in the failure cases).

Predicted Scene Boundaries. To validate the quality of boundary detection using the fine-tuned model, Figure 4 illustrates the success and failure scene boundary predictions of different models. Compared with the state-of-the-art supervised and self-supervised methods, we observe that our CAT can provide better results for scene segmentation. On the other hand, we observe the over-segmentation or under-segmentation issue in several cases (including CAT), but the wrong predictions of CAT are always located close to the ground-truths.

References

Baraldi, L.; Grana, C.; and Cucchiara, R. 2015. A Deep Siamese Network for Scene Detection in Broadcast Videos. In *MM*, 1199–1202. Brisbane, Australia.

Chasanis, V.; Likas, A.; and Galatsanos, N. P. 2009. Scene Detection in Videos Using Shot Clustering and Sequence Alignment. *IEEE Trans. Multim.*, 11(1): 89–100.

Chen, S.; Nie, X.; Fan, D.; Zhang, D.; Bhat, V.; and Hamid, R. 2021. Shot Contrastive Self-Supervised Learning for Scene Boundary Detection. In *CVPR*, 9796–9805. virtual.

Han, B.; and Wu, W. 2011. Video scene segmentation using a novel boundary evaluation criterion and dynamic programming. In *ICME*, 1–6. Catalonia, Spain.

Hendrycks, D.; and Gimpel, K. 2016. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *CoRR*, abs/1606.08415.

Huang, Q.; Xiong, Y.; Rao, A.; Wang, J.; and Lin, D. 2020. MovieNet: A Holistic Dataset for Movie Understanding. In *ECCV*, volume 12349 of *Lecture Notes in Computer Science*, 709–727. Glasgow, UK.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*. San Diego, CA.

Mun, J.; Shin, M.; Han, G.; Lee, S.; Ha, S.; Lee, J.; and Kim, E. 2022. Boundary-aware Self-supervised Learning for Video Scene Segmentation. *CoRR*, abs/2201.05277.

Rao, A.; Xu, L.; Xiong, Y.; Xu, G.; Huang, Q.; Zhou, B.; and Lin, D. 2020. A Local-to-Global Approach to Multi-Modal Movie Scene Segmentation. In *CVPR*, 10143–10152. Seattle, WA.

Rasheed, Z.; and Shah, M. 2005. Detection and representation of scenes in videos. *IEEE Trans. Multim.*, 7(6): 1097–1105.

Rotman, D.; Porat, D.; and Ashour, G. 2017. Optimal Sequential Grouping for Robust Video Scene Detection Using Multiple Modalities. *Int. J. Semantic Comput.*, 11(2): 193–208.

Tapaswi, M.; Bäuml, M.; and Stiefelhagen, R. 2014. StoryGraphs: Visualizing Character Interactions as a Timeline. In *CVPR*, 827–834. Columbus, OH.

Wu, H.; Chen, K.; Luo, Y.; Qiao, R.; Ren, B.; Liu, H.; Xie, W.; and Shen, L. 2022. Scene Consistency Representation Learning for Video Scene Segmentation. *CoRR*, abs/2205.05487.