

# Pseudo Dataset Generation for Out-of-domain Multi-Camera View Recommendation

Kuan-Ying Lee<sup>1</sup>, Qian Zhou<sup>2</sup>, Klara Nahrstedt<sup>1</sup>

<sup>1</sup>*University of Illinois, Urbana-Champaign*

<sup>2</sup>*City University of Hong Kong*

kylee5@illinois.edu, qiazhou@cityu.edu.hk, klara@illinois.edu

**Abstract**—Multi-camera systems are indispensable in movies, TV shows, and other media. Selecting the appropriate camera at every timestamp has a decisive impact on production quality and audience preferences. Learning-based view recommendation frameworks can assist professionals in decision-making. However, they often struggle outside of their training domains. The scarcity of labeled multi-camera view recommendation datasets exacerbates the issue. Based on the insight that many videos are edited from the original multi-camera videos, we propose transforming regular videos into pseudo-labeled multi-camera view recommendation datasets. Promisingly, by training the model on pseudo-labeled datasets stemming from videos in the target domain, we achieve a 68% relative improvement in the model’s accuracy in the target domain and bridge the accuracy gap between in-domain and never-before-seen domains.

**Index Terms**—cinematography, semi-supervised learning

## I. INTRODUCTION

Multi-camera systems capturing the same scene provide different viewing perspectives and play significant roles in movies, broadcasts, news shows, etc. [15]. However, to benefit from multi-camera systems, expertise from cinematography professionals is heavily required. For instance, a video editor has to spend several hours watching pre-recorded videos from all cameras, determining which portions to use, and editing a single-track movie that best narrates the story.

Learning-based multi-camera view recommendation could assist professionals in choosing which camera to switch to by analyzing the past frames that have been selected, thus improving their efficiency. To facilitate the development of learning-based multi-camera recommendation, Rao et al. propose a dataset called TV shows Multi-camera Editing Dataset (TVMCE) [12], providing subsampled frames of multi-camera videos and the camera transitions determined by professional videographers.

However, videos in TVMCE are limited to a few scenes (stages, stadiums, and concerts) and specific types (broadcast or live stream). We found that a multi-camera view recommendation model trained on TVMCE has issues generalizing to never-before-seen domains, and its accuracy drops significantly (31.83 vs. 22.65) when applied to other video scenes (e.g., living rooms) and/or types (e.g., movies).

Collecting data from the same domain would be the most direct approach to solving the issue. Yet, such requires multiple synchronized cameras capturing the same event, not to

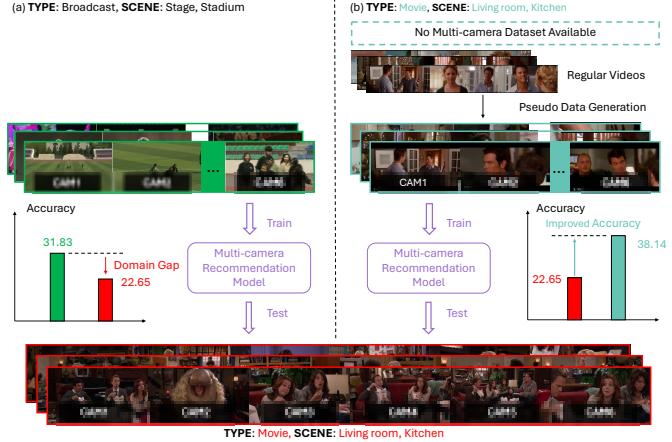


Fig. 1: (a) A model trained on a labeled multi-camera editing dataset of a particular domain generalizes poorly to a never-before-seen domain and the accuracy drops significantly. (b) Our proposed method leverages regular videos to generate pseudo-labeled datasets for the target domain and improve the model’s accuracy. [Best viewed in color.]

mention the dedicated cinematography expertise required for labeling.

This paper proposes a methodology for generating pseudo-labeled multi-camera editing data from regular videos with shots, alleviating data scarcity. With the proposed approach, sufficient data on a given domain (e.g., movie scenes in living rooms) could be obtained. Our insights stem from two observations. (1) Many shot transitions within a regular video result from camera switches<sup>1</sup>. Namely, videos are edited from their original multi-camera videos. (2) In a multiple-camera system, cameras often remain stationary (extrinsics and intrinsics) and are usually responsible for shots of different scales. Based on these two insights, we perform clustering on shots in a video to simulate different cameras and select the most visually similar shot from each camera as candidates alongside the ground truth to generate pseudo-labeled data. A model trained on the proposed dataset enjoys a significant improvement in accuracy in the target domain (22.65 vs. 38.14, cf. Table 3). **Our contributions** are summarized as follows:

<sup>1</sup>Other transitions such as video effect and trimming could be filtered by heuristics or treated as noises.

- We identify the poor domain generalizability of multi-camera view recommendation models.
- We propose generating pseudo-labeled multi-camera editing datasets with regular videos to mitigate the lack of labeled data on an arbitrary domain.
- With the proposed pseudo-labeled multi-camera editing datasets, we achieve a 68% relative improvement in the model’s classification accuracy in the target domain. (cf. Table 3).

## II. RELATED WORK

Many works have studied multi-camera view recommendation [2], [3], [8], [11], [12], [19]. [19] designs two modules that collaboratively make switching decisions on a soccer game based on heuristics like object proximity and view duration. [11] detects pre-defined events of interest in soccer games and leverages a scheduler to decide which view to broadcast. These works design heuristics for a specific event or scene, limiting their applicability to other domains. [8] trains a reinforcement agent to predict video attributes for retrieving the most appropriate view. Similar to our work, [3] leverages off-the-shelf videos as an auxiliary to augment the labeled dataset. The works mentioned above, despite promising, require labeled multi-camera datasets in the target domain. [12] puts up a larger-scale multi-camera editing dataset to foster the growth of the sphere. Yet, the dataset is still limited to particular video styles and cannot be generalized to other domains. Our work differs from the previous works in that we propose a method to transform regular videos of an arbitrary domain into pseudo-labeled multi-camera editing datasets and *altogether bypass the need for human-labeled datasets in the target domain*.

## III. FRAMEWORK

**Problem Setup.** This work aims to assist professionals in video editing by recommending camera tracks to switch to, considering the past shots that the professional has selected. Note that we only focus on which track to switch to, but not whether to switch tracks because multi-camera view recommendation (1) is still nascent and (2) involves a certain amount of subjectivity.

The task is formulated as follows:  $N$  temporally synchronized cameras capture the same scene from different angles, producing  $N$  video tracks. Given a portion of the past video that has been edited from time  $T = s$  to  $T = e$  of length  $e - s$ , the task is to decide which track to switch to at the coming time  $T = t$ , where  $t - e$  could be variable.

As shown in Fig. 2(a), 16 sub-sampled past selected frames (one out of every five frames) are given as input to the model. The output is which of the six cameras should be switched to  $N_F$  frames away. Note that  $N_F$  could be variable.

**Model Architecture.** As shown in Fig. 2(a), the model consists of two main modules: a *feature extractor* encoding individual frames and their corresponding metadata into a single vector, and a *past encoder* aggregating all past features into a learnable latent vector through layer(s) of self-attention. This latent vector encapsulates holistic information from all

the selected frames in the near past, such as visual cues and transitions between past frames. In Fig. 2(b), the feature extractor has two inputs: the frame and its corresponding frame offset to the candidate frames. A Swin Transformer [9] pre-trained on ImageNet [5] encodes the image into an image vector. A positional embedding encodes how distant the input frame is from the candidate frames. For instance, in Fig. 2(b), the frame offset between the most recent past frame and the candidate frames is  $(327 - 299) = 28$ . We use sine and cosine functions of different frequencies as the positional embeddings [17]. We only train the past encoder and the latent vector.

**Training Objectives.** Following previous work on self-supervised feature learning [4], [6], we optimize the model by InfoNCE [10] and maximize the cosine similarity between the past feature and the ground-truth candidate feature while minimizing the cosine similarity between the past feature and all the other candidate features.

## IV. PSEUDO MULTI-CAMERA EDITING DATASET

As mentioned in Sec. I to deal with the scarcity of labeled multi-camera editing data, we propose transforming regular videos in the target domain to pseudo-labeled multi-camera editing data. This section demonstrates the detailed procedure of transforming regular videos into pseudo-labeled multi-camera editing datasets. Fig. 3 illustrates creating a pseudo dataset. First, we detect shots in a video and create pseudo-cameras by clustering the shots. Finally, we select candidates from each pseudo-camera (cluster) to construct an instance.

**Camera Switch.** Our insight stems from the observation that when a professional is editing multi-camera videos, a shot boundary is when they decide to switch from one camera to another. Though we cannot access the original multi-camera videos, the final video carries partial supervision from which the model could learn. We first perform shot detection on a given video and obtain  $N_S$  shots.

**Pseudo Camera Label Generation.** We made two observations. First, in a multi-camera system, each camera tends to remain stationary and keeps its perspective. Second, cameras are usually responsible for shots of different scales. Utilizing the insight that *shot scale of each camera tends to stay unchanged*, we train a Temporal Segment Network (TSN) [18] for shot type classification on MovieShots [13] to extract features for each shot. Then, we obtain six clusters by running K-Means on the features. Each cluster is treated as a pseudo-camera.

**Pseudo Instance Construction.** After assigning a pseudo-camera label to every video shot, we select the most visually similar shot from each pseudo-camera as the candidates alongside the ground truth, as in Fig. 3. We use a ResNet50 [7] pre-trained on ImageNet [5] to extract image features for the first and last frames in each shot and normalize the features to unit length. Then, we compute the cosine similarity of the previous shot’s last-frame feature to the next shot’s first-frame feature as the visual similarity between the two shots.

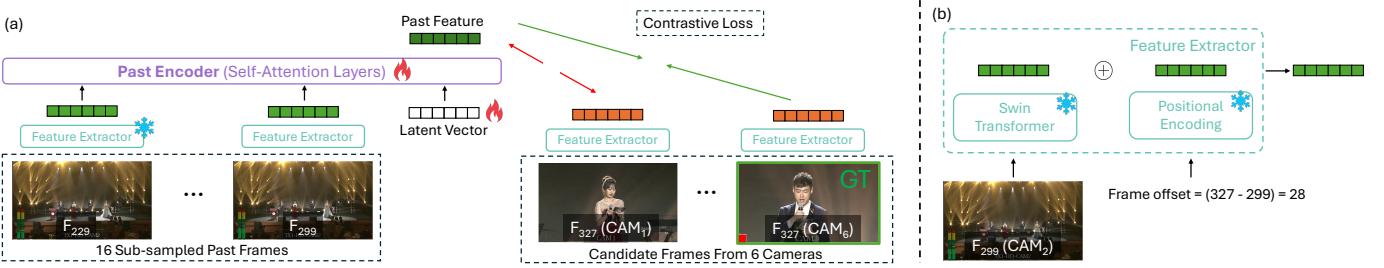


Fig. 2: Model Architecture. (a) The past encoder encodes all past features to a single feature vector. Then, a contrastive loss is applied to maximize the cosine similarity between the past and ground-truth features. (b) The feature extractor encodes a frame by adding a positional embedding to the image feature.  $\text{🔥}$ : trainable,  $\text{❄️}$ : frozen. GT: groundtruth. The number  $F_N$  indicates the  $N^{\text{th}}$  frame from the video. [Best viewed in color.]

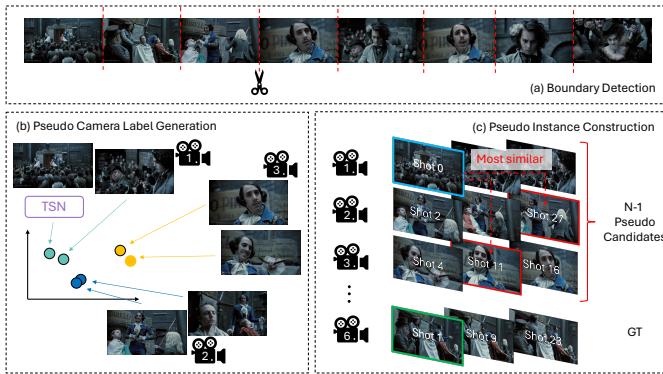


Fig. 3: **Pseudo Dataset Generation Pipeline.** (a) Shots are detected in the input video, and (b) clustered into groups. Shots within the same cluster are regarded as from the same “pseudo” camera. (c) A shot is selected as an **anchor**. The succeeding shot is the **ground truth**, while the most similar shot amongst each of the other  $N-1$  pseudo cameras is chosen as a **candidate**. [Best viewed in color with zoom-in.]

## V. EXPERIMENT

### A. Setup

**Datasets.** We leverage four datasets: **TVMCE** [12] and three pseudo datasets created from **ClipShots** [16], **Condensed Movies** [11], and **Sitcoms** episodes. **TVMCE** consists of 6 synchronized camera tracks, totaling 88 hours of recorded videos. The professionals edit the multi-camera videos into a single-track video, providing 5133 ground-truth camera transition labels. Following [12], 4042 and 1091 transitions are used for training and testing, respectively. **ClipShots** [16] is a dataset for shot boundary detection. It contains videos from over 20 categories, ranging widely from movie spotlights to phone videos. After removing shots with gradual transitions, e.g., fade, we leverage the remaining shots for pseudo-dataset generation. In total, there are 118658 camera transitions. **Condensed Movies** [11] is proposed to understand the narrative structure of movies and contains key scenes from over 3K videos, providing 152893 camera transitions. Shots from one episode of each of the four **Sitcoms**: Friends, How I Met Your

Method↓ / Test Set→	TVMCE (ID)	Sitcoms (OOD)
Random	16.67	16.67
TC Transformer	28.77±0.86	14.04±1.59
<b>Ours</b>	<b>31.83±0.83</b>	<b>22.65±2.43</b>

Table 1: **Comparison to Baseline.** Multi-camera view recommendation models generalize poorly to a never-before-seen domain. ID: in-Domain, OOD: out-of-Domain.

Mother, The Big Bang Theory, and Two and a Half Men are used for pseudo dataset generation.

**Implementation Details.** All models are trained for ten epochs, with a learning rate of 1e-5 and a batch size of 2. The mean and standard deviation of results from three seeds are reported for each setup. TransNet V2 [14] is used for shot detection. TSN pre-trained on MovieShots [13] for 60 epochs, reaching a top-1 shot scale classification accuracy of 90.08%, is used for feature extraction. Specifically, features from the penultimate layer are used for shot clustering. We remove videos with fewer than ten shot transitions, as transitions in these videos are likely from video effects but not actual camera transitions. We also discard shots with gradual transitions. For evaluation, **Classification accuracy** of the camera being switched to is used.

### B. Result

**Baseline Comparison.** A few works have been proposed for multi-camera view recommendation [3], [8], [11]. Yet, neither their codes nor their datasets are publicly available. We re-implement the Temporal and Contextual Transformer (TC Transformer) in [12] as the primary baseline we compare to. It achieves state-of-the-art (SoTA) classification accuracy in multi-camera view recommendation on TVMCE dataset. Note that we remove the training data in the middle of a shot (without camera switches), which improves the accuracy (22.48 in [12]) of TC Transformer to 28.77. As shown in Table 1, the proposed framework outperforms TC Transformer by 11% (31.83 vs. 28.77). However, both models suffer when applied to a never-before-seen domain.

TVMCE test set — video scenes: <b>stage, concert hall</b>		
Train Set	Video Scene	Accuracy
TVMCE_stage	<b>stage, concert hall</b>	31.83 $\pm$ 0.83
TVMCE_sport	<b>sports</b>	25.56 $\pm$ 0.87
ClipShots ( <b>Pseudo</b> )	<b>broad</b> <sup>1</sup>	26.81 $\pm$ 0.22

<sup>1</sup>consists of videos from a broad scene coverage.

Table 2: **Impact of Video Scene.** The model trained in different scenes to the test set achieves lower accuracy. **Green**, and **red** means same and different. [Best viewed in color.]

**Generalizability & Pseudo Dataset Efficacy.** We investigate domain differences from two perspectives: (1) *video scene*, and (2) *video type*. Video scene is where the videos are filmed, e.g., on stage or in a livingroom. Video type is how the contents are presented, e.g., a live performance or a movie.

First, we fix video type and examine the impact of video scene. Concretely, we split the TVMCE training set into two roughly equal-sized subsets based on the video scenes. One subset, **TVMCE\_stage**, similar to the test set, consists of videos of stages and concert halls. The other subset, **TVMCE\_sport**, contains sports videos. In Table 2, the model trained on the dissimilar scenes achieves significantly lower accuracy than one trained on the same scenes as the test set.

Next, we investigate the impact of both video scenes and video types. In this experiment, all models are evaluated on **Sitcoms**, consisting of videos that differ in both scenes and types from those in TVMCE. Specifically, video scenes in TVMCE are mostly stages and sports, whereas video scenes in Sitcoms are kitchens and living rooms. Also, TVMCE contains recorded live performances, whereas Sitcoms contains movies. Three models are evaluated, each trained on a dataset with different degrees of domain difference to Sitcoms. (1) TVMCE that differs in both video type and scenes, (2) ClipShots differs only in video type, and (3) Condensed Movies that covers the same video type and scenes. Two observations can be made in Table 3. (1) Pseudo datasets bridge the data scarcity and improve the accuracy in the target domain. (2) Video type and scene can compound to impact performance. TVMCE, different in both video type and scenes from Sitcoms, performs the worst. On the contrary, Condensed Movies with the same video type and scenes as Sitcoms achieves the best performance.

Better accuracy of the model trained on pseudo multi-camera editing datasets — ClipShots in Table 2, and ClipShots and Condensed Movies in Table 3 demonstrate their efficacy in improving model performance in the target domain without labeling by professionals.

Note that previous work also leverages pseudo labels for evaluation [8]; still, an important question is whether pseudo-labeled datasets reflect the performance of human-labeled ones. We treat the TVMCE test set as unlabeled, construct a pseudo-TVMCE test set, and evaluate the model trained with sports scenes on this set. The accuracy is similar to the TVMCE test set (23.71 $\pm$ 0.12 vs. 25.56 $\pm$ 0.87).

**Pseudo Dataset Construction Methodology.** Three method-

Test set — video type: <b>movie</b> , video scenes: <b>living room, kitchen</b>			
Train Set	Video Type	Video Scene	Accuracy
TVMCE	<b>live</b>	<b>stages, sports</b>	22.65 $\pm$ 2.43
ClipShots ( <b>Pseudo</b> )	<b>mixed</b> <sup>1</sup>	<b>broad</b> <sup>2</sup>	27.61 $\pm$ 1.20
Condensed ( <b>Pseudo</b> )	<b>movie</b>	<b>broad</b> <sup>2</sup>	<b>38.14<math>\pm</math>1.50</b>

<sup>1</sup>consists of videos of movies and live performance.

<sup>2</sup>consists of videos from a broad scene coverage.

Table 3: **Impact of Video Scene and Type (Sitcoms).** A more significant domain difference (video scenes + video types) severely impacts accuracy. Pseudo datasets from a broad range of videos could cover the target video scenes and types, achieving better accuracy. **Green**, **orange**, and **red** means same, covered, and different. [Best viewed in color.]

ologies are investigated: (1) selecting the most visually similar shot from each cluster, (2) selecting one random shot from each cluster, and (3) selecting the five most similar shots *without clustering*. On the TVMCE test set, the three models achieve an accuracy of 26.81 $\pm$ 0.22, 7.16 $\pm$ 0.40, and 26.02 $\pm$ 0.49, respectively. The discrepancy between (1) and (2) shows the importance of selecting the shots that are most visually similar to the candidates. We conjecture that the ground-truth succeeding shots tend to have a certain amount of overlap in viewpoint with their predecessors, and randomly selected shots with little visual overlap to the previous shot could be quickly ruled out by the model, leading to ineffective training. The difference between (1) and (3) shows that while clustering cannot recover the actual cameras, it can be used to obtain hard negatives, thus facilitating model training.

## VI. DISCUSSION

**Application Scenarios.** Based on the experiment results, we divide multi-camera view recommendation into three scenarios. (1) In a controlled environment where decisions made by professionals for similar events are accessible. For example, in soccer games, the camera transition patterns are relatively straightforward, and multi-camera videos are available alongside the final broadcast. In this scenario, supervised learning on the professional-labeled datasets would suffice. (2) If the model were to be applied to domains where decisions made previously by experts are unavailable, we would suggest generating pseudo datasets on the target domains with off-the-shelf regular videos for training. (3) If one does not know the explicit target domains and wants to apply the model in the wild, the best strategy would be to collect videos from extensive and different domains and train the model on pseudo datasets transformed from these videos. This would increase the probability of the pseudo datasets covering the target domains. We note that the proposed pseudo multi-camera editing dataset *does not solve the innate domain gap issue* once and for all but mitigates it through the availability of regular videos in an arbitrary domain.

## VII. CONCLUSION

In this paper, we first show that multi-camera view recommendation models struggle to generalize to never-before-seen domains. We then analyze two aspects that could compound and intensify domain mismatch: video scenes and video types. We propose to leverage regular videos in the target domain to generate pseudo multi-camera editing datasets. We also design a learning framework that optimizes the model parameters with contrastive loss to bring close the current and the succeeding ground-truth shots. Experiments demonstrate the efficacy of the proposed pseudo multi-camera editing dataset in improving the model’s accuracy in the target domain.

Future work involves designing a better training objective to capture the underlying cinematography expertise, which enables professionals to work across video scenes and types. Another potential direction is to incorporate explicit rules of thumb and conventions in cinematography, e.g., the gradual change from wider to narrower shots and the rule of thirds.

## VIII. ACKNOWLEDGEMENT

This work was funded by National Science Foundation grants NSF CNS 19-00875, NSF CCF 22-17144, NSF CNS 21-06592. Any results and opinions are our own and do not represent views of National Science Foundation.

## REFERENCES

- [1] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings. In *ACCV*, 2020.
- [2] Saugata Biswas, Ernst Kruijff, and Eduardo Veas. View recommendation for multi-camera demonstration-based training. *Multimedia Tools and Applications*, 83(7):21765–21800, 2024.
- [3] Jianhui Chen, Keyu Lu, Sijia Tian, and Jim Little. Learning sports camera selection from internet videos. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1682–1691. IEEE, 2019.
- [4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.
- [6] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [8] Panwen Hu, Nan Xiao, Feifei Li, Yongquan Chen, and Rui Huang. A reinforcement learning-based automatic video editing method using pre-trained vision-language model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 6441–6450, 2023.
- [9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.
- [10] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [11] Yingwei Pan, Yue Chen, Qian Bao, Ning Zhang, Ting Yao, Jingren Liu, and Tao Mei. Smart director: An event-driven directing system for live broadcasting. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(4):1–18, 2021.
- [12] Anyi Rao, Xuekun Jiang, Sichen Wang, Yuwei Guo, Zihao Liu, Bo Dai, Long Pang, Xiaoyu Wu, Dahua Lin, and Libiao Jin. Temporal and contextual transformer for multi-camera editing of tv shows. *arXiv preprint arXiv:2210.08737*, 2022.
- [13] Anyi Rao, Jiaze Wang, Lining Xu, Xuekun Jiang, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A unified framework for shot type classification based on subject centric lens. In *ECCV*, pages 17–34. Springer, 2020.
- [14] Tomáš Souček and Jakub Lokoč. Transnet v2: An effective deep network architecture for fast shot transition detection. *arXiv preprint arXiv:2008.04838*, 2020.
- [15] Eckhard Stoll, Stephan Breide, Steve Göring, and Alexander Raake. Automatic camera selection, shot size and video editing in theater multi-camera recordings. *IEEE Access*, 2023.
- [16] Shitao Tang, Litong Feng, Zhanghui Kuang, Yimin Chen, and Wei Zhang. Fast video shot transition localization with deep structured models. In *ACCV*, pages 577–592. Springer, 2018.
- [17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 30, 2017.
- [18] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016.
- [19] Xueting Wang, Yuki Muramatu, Takatsugu Hirayama, and Kenji Mase. Context-dependent viewpoint sequence recommendation system for multi-view video. In *2014 IEEE International Symposium on Multimedia*, pages 195–202. IEEE, 2014.