BOUNDARY-AWARE SELF-SUPERVISED LEARNING FOR VIDEO SCENE SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised learning has drawn attention through its effectiveness in learning in-domain representations with no ground-truth annotations; in particular, it is shown that properly designed pretext tasks (e.g., contrastive prediction task) bring significant performance gains for a downstream task (e.g., classification task). Inspired from this, we tackle video scene segmentation, which is a task of temporally localizing scene boundaries in a video, with a self-supervised learning framework where we mainly focus on designing effective pretext tasks. In our framework, we discover a pseudo-boundary from a sequence of shots by splitting it into two continuous, non-overlapping sub-sequences and leverage the pseudo-boundary to facilitate the pre-training. Based on this, we introduce three novel boundary-aware pretext tasks: 1) Shot-Scene Matching (SSM), 2) Contextual Group Matching (CGM) and 3) Pseudo-boundary Prediction (PP); SSM and CGM guide the model to maximize intra-scene similarity and inter-scene discrimination while PP encourages the model to identify transitional moments. Through comprehensive analysis, we empirically show that pre-training and transferring contextual representation are both critical to improving the video scene segmentation performance. Lastly, we achieve the new state-of-the-art on the MovieNet-SSeg benchmark. The code will be released.

1 INTRODUCTION

The video scene segmentation is a task of identifying scene boundaries from a video where a scene is defined as a semantic unit for making a story and is composed of a series of semantically cohesive shots—a set of frames captured by the same camera during an uninterrupted period of time—in the same context. Localizing scene boundaries is a significant step towards the high-level video understanding because dividing a long video into a set of meaningful scenes enables models to inspect the individual incidents from complex story.

One of the biggest challenges with temporal semantic segmentation is that it is not achieved simply by detecting changes in visual cues. As shown in Figure 1(a), we present an example of nine shots, all of which belong to the same scene, where two characters are talking on the phone. We can see that the overall visual cues within the scene do not stay the same but rather change repeatedly when each character appears. On the other hand, as presented in Figure 1(b), the other example shows two different scenes which contain visually similar shots (highlighted in blue) where the same character appears in the same place. Therefore, it is expected that two adjacent scenes which share shots with similar visual cues need to be contextually discriminated. From this observation, it is important for the video scene segmentation task to model contextual relation between shots by maximizing 1) *intra-scene similarity* (*i.e.*, the shots belonging to the same scene should be close to each other), and 2) *inter-scene discrimination* across two adjacent scenes (*i.e.*, two neighbor shots across the scene boundary should be distinguishable).

Supervised learning approaches (*e.g.*, Rao et al. (2020)) are clearly limited due to the lack of large-scale datasets with reliable ground-truth annotations. Recently, self-supervision (Chen et al., 2020a; Caron et al., 2020; He et al., 2020; Roh et al., 2021) is spotlighted through its effectiveness in learning in-domain representation without relying on costly ground truth annotations. The self-supervised learning methods (Chen et al., 2021; Feichtenhofer et al., 2021; Dave et al., 2021; Qian et al., 2021) in the video domain are often designed to learn spatio-temporal patterns in short clips

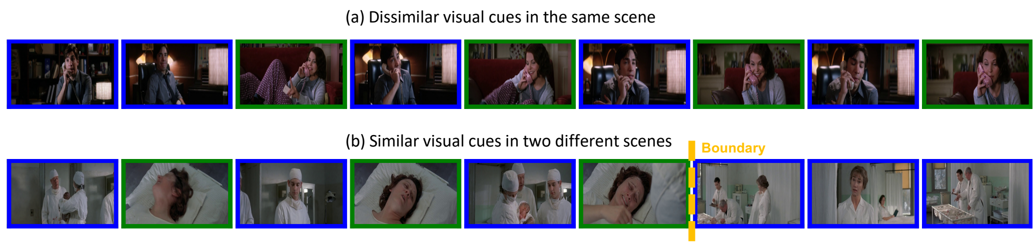

Figure 1: Examples of the video scene segmentation. In each row, we visualize the shots including similar visual cues (*e.g.*, characters, places, etc.) with the same colored border.

(*e.g.*, shots in movies). This kind of learned representation is generic and can be applied to many video understanding tasks (*e.g.*, action classification). However, such representation is not sufficient for video scene segmentation because this task requires not only a good representation for individual shots but also contextual representation considering neighboring shots at a higher level as illustrated in Figure 1. Motivated by this, we set our main goal to design effective self-supervised objectives (*i.e.*, pretext tasks) that maximize intra-scene similarity as well as discriminate shots from different scenes. For the purpose, this raises a penetrating question: *how can we design boundary-relevant pretext tasks without access to ground truth boundary annotations?*

We introduce a novel **B**oundary-**a**ware **S**elf-**S**upervised **L**earning (BaSSL) framework. Our main idea of BaSSL is to localize a pseudo-boundary, which is obtained by dividing the input sequence of shots into two semantically disjoint sub-sequences, and use it to define pretext tasks that are beneficial to the video scene segmentation task. On top of the discovered two sub-sequences and a pseudo-boundary, three boundary-aware pretext tasks are proposed: 1) Shot-Scene Matching (SSM); 2) Contextual Group Matching (CGM); and 3) Pseudo-boundary Prediction (PP). Note that SSM and CGM encourage the model to maximize intra-scene similarity and inter-scene discrimination while PP enables the model to learn the capability of identifying transitional moments. In addition, we perform Masked Shot Modeling (MSM) task inspired by Sun et al. (2019a) to further learn temporal relationship between shots. The comprehensive analysis demonstrates the effectiveness of the proposed framework (*i.e.*, pre-training of contextual relationship between shots) as well as the contribution of the proposed individual components (*i.e.*, the algorithm for pseudo-boundary discovery and boundary-aware pretext tasks).

Our main contributions are summarized as follows: (i) We introduce a novel boundary-aware pre-training framework which adopts dynamic time warping (DTW) algorithm to identify pseudo-boundaries and use them as self-supervision to facilitate the pre-training; (ii) we propose three boundary-aware pretext tasks, which are carefully designed to learn essential capabilities required for the video scene segmentation task; (iii) we perform extensive ablations to demonstrate the effectiveness of the proposed framework, including the observation that our framework is complementary to the existing framework; (iv) we achieve the new state-of-the-art on the MovieNet-SSeg benchmark while outperforming existing self-supervised learning-based methods by large margins.

2 Related Work

Video scene segmentation approaches formulate the task as a problem of temporal grouping of shots. In this formulation, the optimal grouping can be achieved by clustering-based (Rui et al., 1998; Rasheed & Shah, 2003; 2005; Chasanis et al., 2008), dynamic programming-based (Han & Wu, 2011; Tapaswi et al., 2014; Rotman et al., 2017) or multi-modal input-based (Liang et al., 2009; Sidiropoulos et al., 2011) methods. However, the aforementioned methods have been trained and evaluated on small-scale datasets such as OVSD (Rotman et al., 2016) and BBC (Baraldi et al., 2015) which can produce a poorly generalized model. Recently, Huang et al. (2020) introduce a large-scale video scene segmentation dataset (*i.e.*, MovieNet-SSeg) that contains hundreds of movies. Training with large-scale data, Rao et al. (2020) proposes a strong supervised baseline model that performs a shot-level binary classification followed by grouping using the prediction scores. In addition, Chen et al. (2021) proposes a shot contrastive pre-training method that learns shot-level representation. We found ShotCoL (Chen et al., 2021) to be the most similar work to our method. However, our method is different from ShotCoL in that we specifically focus on learning contextual representations by considering the relationship between shots. We refer interested readers to the supplementary material for a more detailed analysis of this.

Action segmentation in videos is one of the related works for video scene segmentation, which identifies action labels of individual frames, thus divides a video into a series of action segments. Supervised methods (Lea et al., 2016; Farha & Gall, 2019) proposed CNN-based architectures to effectively capture temporal relationship between frames in order to address an over-segmentation issue. As frame-level annotations are prohibitively costly to acquire, weakly supervised methods (Chang et al., 2019; Li et al., 2019; Li & Todorovic, 2020; Souri et al., 2021; Shen et al., 2021; Zhukov et al., 2019; Fried et al., 2020) have been suggested to use an ordered list of actions occurring in a video as supervision. Most of the methods are trained to find (temporal) semantic alignment between frames and a given action list using an HMM-based architecture (Kuehne et al., 2018), a dynamic programming-based assignment algorithm (Fried et al., 2020) or a DTW-based temporal alignment method (Chang et al., 2019). Recently, unsupervised methods (Kumar et al., 2021; Wang et al., 2021; Kukleva et al., 2019; Li & Todorovic, 2021; VidalMata et al., 2021) have been further proposed; in a nutshell, clustering-based prototypes are discovered from unlabeled videos, then the methods segment the videos by assigning prototypes (corresponding to one of the actions) into frames. In contrast to the action segmentation task that is limited to localizing segments each of which represents a single action within an activity, video scene segmentation requires localizing more complex segments each of which may be composed of more than two actions (or activities).

Self-supervised learning in videos has been actively studied for the recent years with approaches proposing various pretext tasks such as future frame prediction (Srivastava et al., 2015; Vondrick et al., 2016; Ahsan et al., 2018), temporal ordering of frames (Misra et al., 2016; Lee et al., 2017; Xu et al., 2019), geometric transformations prediction (Jing & Tian, 2018), colorization of videos (Vondrick et al., 2018) and contrastive prediction (Feichtenhofer et al., 2021; Qian et al., 2021; Dave et al., 2021). In addition, CBT (Sun et al., 2019a;b) proposes a pretext task of masked frame modeling to learn temporal dependency between frames (or clips). Note that since most of those methods are proposed for the classification task, they would be sub-optimal to the video scene segmentation task. On the other hand, BSP (Xu et al., 2020) proposes boundary-sensitive pre-text tasks based on synthesized pseudo-boundaries that are obtained by concatenating two clips sampled from different videos. However, strictly speaking, BSP is not a self-supervised learning algorithm since it requires video-level class labels to synthesize pseudo-boundaries; the proposed pretext tasks are not applicable to videos such as movies that are hard to define semantic labels. Also, note that we empirically show that pseudo-boundaries identified by our method are more effective for pre-training than synthesized pseudo-boundaries.

3 BOUNDARY-AWARE SELF-SUPERVISED LEARNING (BASSL)

In this section, we introduce our proposed approach, Boundary-aware Self-Supervised Learning (BaSSL). We start with the problem formulation followed by the model overview. Then, we describe our novel boundary-aware pretext tasks for pre-training.

3.1 PROBLEM FORMULATION

Terminologies A video (*e.g.*, documentaries, TV episodes and movies) is a sequence of scenes, defined as a semantic unit for making a story. A scene is a series of shots, which is a set of frames physically captured by the same camera during an uninterrupted period of time.

Problem Definition Given a video, which contains a series of N shots $\{\mathbf{s}_1, ..., \mathbf{s}_N\}$ with class labels $\{y_1, ..., y_N\}$ where $y_i \in \{0, 1\}$ indicating if it is at the scene boundary (more precisely, if it is the last shot of a scene), the video scene segmentation task is formulated as a simple binary classification problem at individual shot level. By definition, a scene boundary is where the semantic of a shot is considerably different from its (one-way) neighbors. Thus, it is in nature important to capture and leverage contextual transition across the scenes. Consequently, it is a common practice that the information of the neighbor shots are leveraged together when determining scene boundaries. With this formulation, existing supervised learning approaches typically train a parameterized (θ) model by maximizing the expected log-likelihood:

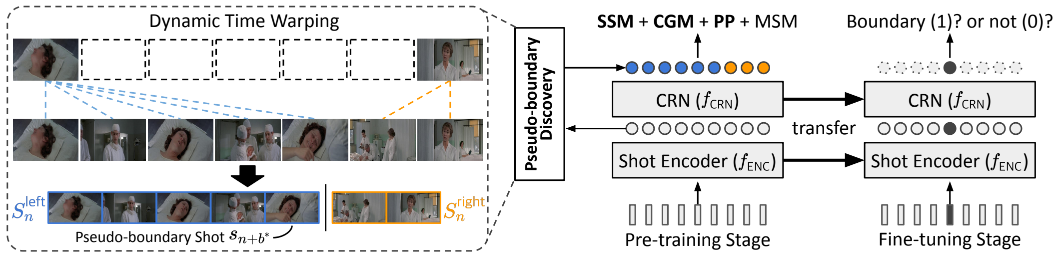

Figure 2: Overall pipeline of our proposed framework, BaSSL.

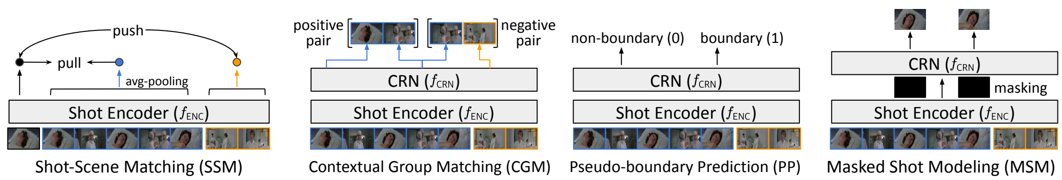

Figure 3: Illustration of four pre-training pretext tasks.

$$\theta^* = \arg\max_{\theta} \mathbb{E}\left[\log p_\theta(y_n|\mathbf{S}_n)\right], \tag{1}$$

where $\mathbf{S}_n = \{\mathbf{s}_{n-K}, ..., \mathbf{s}_n, ..., \mathbf{s}_{n+K}\}$ is a set of $2K+1$ shots centered at n^{th} shot $\mathbf{s}_n$, and K is the number of neighbor shots before and after $\mathbf{s}_n$. Note that each shot $\mathbf{s}$ is given by a set of N_k key-frames, resulting in a tensor with size of (N_k, C, H, W) where C, H and W are the RGB channels, the frame height and the frame width, respectively.

3.2 Model Overview

Our method is based on two-stage training following common practice (Chen et al., 2021): pre-training on large-scale unlabeled data with self-supervision and fine-tuning on relatively small labeled data via transfer learning. Our main focus is in the pre-training stage, aiming at designing effective pretext tasks for video scene segmentation.

As illustrated in Figure 2, our model consists of two main components: 1) *shot encoder* embedding a shot by capturing its spatio-temporal patterns, and 2) *contextual relation network* (CRN) capturing relationship between shots. Given a sequence $\mathbf{S}_n = \{\mathbf{s}_{n-K}, ..., \mathbf{s}_n, ..., \mathbf{s}_{n+K}\}$ of shots centered at $\mathbf{s}_n$, two-level representations are extracted as follows:

$$\mathbf{e}_n = f_{\text{ENC}}(\mathbf{s}_n) \;\text{ and }\; \mathbf{C}_n = f_{\text{CRN}}(\mathbf{E}_n), \tag{2}$$

where $f_{\text{ENC}}\colon \mathbb{R}^{N_k\times C\times H\times W} \to \mathbb{R}^{D_e}$ and $f_{\text{CRN}}\colon \mathbb{R}^{(2K+1)\times D_e} \to \mathbb{R}^{(2K+1)\times D_c}$ represent the shot encoder and the contextual relation network while D_e and D_c mean dimensions of encoded and contextualized features, respectively. $\mathbf{e}_n$ is an encoding of shot $\mathbf{s}_n$ by f_{ENC} while $\mathbf{E}_n = \{\mathbf{e}_{n-K}, ..., \mathbf{e}_n, ..., \mathbf{e}_{n+K}\}$ and $\mathbf{C}_n = \{\mathbf{c}_{n-K}, ..., \mathbf{c}_n, ..., \mathbf{c}_{n+K}\}$ correspond to the input and output feature sequence for f_{CRN}, respectively.

On top of encoded shot representations, BaSSL extracts a pseudo-boundary (left box of Figure 2) to self-supervise the model instead of relying on ground-truth annotations. To be specific, we leverage the dynamic time warping technique to divide the input sequence of shots into two semantically disjoint sub-sequences and output a pseudo-boundary. (See Section 3.3 for more details.)

Then, as presented in Figure 3, using the discovered pseudo-boundary, we devise three novel boundary-aware pretext tasks in Section 3.4: 1) *Shot-Scene Matching* to match shots with their associated scenes, 2) *Contextual Group Matching* to align shots whether they belong to the same scene or not and 3) *Pseudo-boundary Prediction* to capture semantic changes. In addition, we adopt the masked shot modeling in CBT (Sun et al., 2019a) to further learn temporal relationship between shots. After trained with the four pretext tasks, the model is fine-tuned with labeled video scene segmentation data. (See Section 3.5 for more details.)

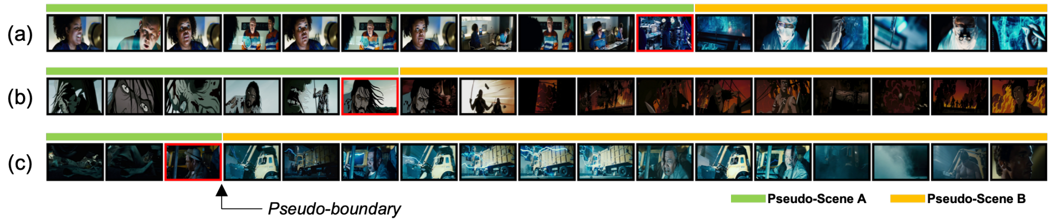

Figure 4: An example in each row shows a sequence of shots sampled from the same scene where there exists no ground-truth scene-level boundary. Our method finds a pseudo-boundary shot (highlighted in red) that divides a sequence into two pseudo-scenes (represented by green and orange bars, respectively) so that semantics (*e.g.*, places, characters) maximally changes.

3.3 PSEUDO-BOUNDARY DISCOVERY

During the pre-training, no scene boundary annotation is available for the input sequence $\mathbf{S}_n$ of shots. Thus, we infer pseudo-boundaries which serve as if they are ground truth labels for boundary-relevant pretext tasks. In addition, we simplify the problem to always extract a single boundary given an input sequence. This is reasonable because even a sequence of shots without strong scene-level semantic transition, there always exists a shot within the sequence across which the semantic transition is maximum, and we use this shot as a pseudo-boundary. To support our claim, we show examples in Figure 4 where we intentionally infer a pseudo-boundary based on a sequence of shots sampled from the same scene (that is, no scene boundary exists between shots according to the ground truth); we observe that the resulting two sub-sequences are cognitively distinguishable.

This simplified problem, dividing the input sequence $\mathbf{S}_n$ into two continuous, non-overlapping sub-sequences $\mathbf{S}_n^{\text{left}}$ and $\mathbf{S}_n^{\text{right}}$, can be seen as a temporal alignment problem between $\mathbf{S}_n$ and $\mathbf{S}_n^{\text{slow}}$; specifically, observing the first shot should belong to $\mathbf{S}_n^{\text{left}}$ and the last one to $\mathbf{S}_n^{\text{right}}$, we define $\mathbf{S}_n^{\text{slow}} = \{\mathbf{s}_{n-K}, \mathbf{s}_{n+K}\}$, which can be seen as a same video with $\mathbf{S}_n$ with lower sampling frequency. Finally, the task becomes aligning intermediate shots either to $\mathbf{S}_n^{\text{left}}$ or $\mathbf{S}_n^{\text{right}}$ preserving continuity.

Under the problem setting, we adopt dynamic time warping (DTW) (Berndt & Clifford, 1994) to find the optimal temporal alignment between $\mathbf{S}_n$ and $\mathbf{S}_n^{\text{slow}}$. In detail, DTW solves the following optimization problem using dynamic programming to maximize semantic coherence of the resulting two sub-sequences among all possible boundary candidates:

$$b^* = \underset{b=-K+1,\ldots,K-1}{\arg\max} \frac{1}{b+K} \sum_{i=-K+1}^{b} \text{sim}(\mathbf{e}_{n-K}, \mathbf{e}_{n+i}) + \frac{1}{K-b-1} \sum_{j=b+1}^{K-1} \text{sim}(\mathbf{e}_{n+K}, \mathbf{e}_{n+j}), \tag{3}$$

where b is the candidate boundary offset, b^* is the optimal boundary offset, and $\text{sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$ computes cosine similarity between encodings of the given two shots. Two sub-sequences are inferred as $\mathbf{S}_n^{\text{left}} = \{\mathbf{s}_{n-K}, \ldots, \mathbf{s}_{n+b^*}\}$ and $\mathbf{S}_n^{\text{right}} = \{\mathbf{s}_{n+b^*+1}, \ldots, \mathbf{s}_{n+K}\}$. $\mathbf{s}_{n+b^*}$ is the pseudo-boundary shot, which is the last shot of $\mathbf{S}_n^{\text{left}}$. More examples of pseudo-boundaries identified by our algorithm is presented in Figure 7 of the supplementary material. The results are used for learning boundary-aware pretext tasks, which will be described below.

3.4 PRE-TRAINING OBJECTIVES

As illustrated in Figure 3, we adopt four pretext tasks for pre-training—1) shot-scene matching, 2) contextual group matching, 3) pseudo-boundary prediction, and 4) masked shot modeling—and the final pre-training loss is defined by

$$\mathcal{L}_{\text{pretrain}} = \mathcal{L}_{\text{SSM}} + \mathcal{L}_{\text{CGM}} + \mathcal{L}_{\text{PP}} + \mathcal{L}_{\text{MSM}}. \tag{4}$$

Shot-Scene Matching (SSM) The objective of this task is to make the representations of a shot and its associated scene similar to each other, while the representations of the shot and other scenes dissimilar. In other words, SSM encourages the model to maximize intra-scene similarity, while minimizing inter-scene similarity. Considering the splitted two sub-sequences ($\mathbf{S}_n^{\text{left}}$ and $\mathbf{S}_n^{\text{right}}$) as pseudo-scenes, we train the model using the InfoNCE loss (Oord et al., 2018):

$$\mathcal{L}_{\text{SSM}} = \mathcal{L}_{\text{NCE}}\left(h_{\text{SSM}}(\mathbf{e}_{n-K}), h_{\text{SSM}}(\mathbf{r}(\mathbf{S}_n^{\text{left}}))\right) + \mathcal{L}_{\text{NCE}}\left(h_{\text{SSM}}(\mathbf{e}_{n+K}), h_{\text{SSM}}(\mathbf{r}(\mathbf{S}_n^{\text{right}}))\right), \quad (5)$$

$$\mathcal{L}_{\text{NCE}}(\mathbf{e}, \mathbf{r}) = -\log \frac{\exp(\text{sim}(\mathbf{e}, \mathbf{r})/\tau)}{\exp(\text{sim}(\mathbf{e}, \mathbf{r})/\tau) + \sum_{\bar{\mathbf{e}} \in \mathcal{N}_e} \exp(\text{sim}(\bar{\mathbf{e}}, \mathbf{r})/\tau) + \sum_{\bar{\mathbf{r}} \in \mathcal{N}_r} \exp(\text{sim}(\mathbf{e}, \bar{\mathbf{r}})/\tau)}, \quad (6)$$

where h_{SSM} is a SSM head of a linear layer, τ is a temperature hyperparameter and $\mathbf{r}(\mathbf{S})$ means a scene-level representation; we use the averaged encoding of shots in the sub-sequence $\mathbf{S}$. $\mathcal{N}_e$ and $\mathcal{N}_r$ in Eq. (6) are constructed using other shots and pseudo-scenes in a mini-batch, respectively.

Contextual Group Matching (CGM) Since directly matching representations of shots and scenes would not be effective when the scenes are composed of visually dissimilar shots, CGM is introduced to bridge this gap. Similar to SSM, CGM is also designed to maximize intra-scene similarity and inter-scene discrimination. However, CGM measures semantic coherence of the shots rather than comparing visual cues. With CGM, the model learns to decide if the given two shots belong to the same group (*i.e.*, scene) or not. In detail, we use the center shot $\mathbf{s}_n$ in the input sequence as the anchor and construct a triplet of ($\mathbf{s}_n$, $\mathbf{s}_{\text{pos}}$, $\mathbf{s}_{\text{neg}}$). We sample each shot from $\mathbf{S}_n^{\text{left}}$ and $\mathbf{S}_n^{\text{right}}$; the one sampled within the same sub-sequence with $\mathbf{s}_n$ is used as the positive shot $\mathbf{s}_{\text{pos}}$, while the other as the negative $\mathbf{s}_{\text{neg}}$. The CGM loss is defined using a binary cross-entropy loss as follows:

$$\mathcal{L}_{\text{CGM}} = -\log\left(h_{\text{CGM}}(\mathbf{c}_n, \mathbf{c}_{\text{pos}})\right) - \log\left(1 - h_{\text{CGM}}(\mathbf{c}_n, \mathbf{c}_{\text{neg}})\right), \quad (7)$$

where h_{CGM} is a CGM head taking two shots as input and predicting a matching score. $\mathbf{c}_n$, $\mathbf{c}_{\text{pos}}$ and $\mathbf{c}_{\text{neg}}$ are the contextualized features by f_{CRN} for the center, positive and negative shots, respectively.

Pseudo-boundary Prediction (PP) Through the above two pretext tasks, our model learns the contextual relationship between shots. In addition to these, we design an extra pretext task, PP, which is more directly related to boundary detection; PP makes the model have a capability of identifying transitional moments that semantic changes. Based on the pseudo-boundary shot and one randomly sampled non-boundary shot, the PP loss is defined as a binary cross-entropy loss:

$$\mathcal{L}_{\text{PP}} = -\log\left(h_{\text{PP}}(\mathbf{c}_{n+b^*})\right) - \log\left(1 - h_{\text{PP}}(\mathbf{c}_{\bar{b}})\right), \quad (8)$$

where h_{PP} is a PP head that projects the contextualized shot representation to a probability distribution over binary class. $\mathbf{c}_{n+b^*}$ and $\mathbf{c}_{\bar{b}}$ indicate the contextualized representation from f_{CRN} for the pseudo-boundary shot $\mathbf{s}_{n+b^*}$ and randomly sampled non-boundary shot $\mathbf{s}_{\bar{b}}$, respectively.

Masked Shot Modeling (MSM) Inspired by masked frame modeling (Sun et al., 2019a;b), we adopt the MSM task whose goal is to reconstruct the representation of masked shots based on the their surrounding shots. In this task, given a set of encoded shot representations, we randomly apply masking each of them with a probability of 15%. For a set $\mathcal{M}$ of masked shot offsets, we learn to regress the output on each masked shot to its encoded shot representation, which is given by

$$\mathcal{L}_{\text{MSM}} = \sum_{m \in \mathcal{M}} \|\mathbf{e}_m - h_{\text{MSM}}(\mathbf{c}_m)\|_2^2, \quad (9)$$

where h_{MSM} is a MSM head to match the dimension of contextualized shot representation with that of encoded one. $\mathbf{e}_m$ and $\mathbf{c}_m$ denote the encoded representation by f_{ENC} and contextualized representation by f_{CRN} for a masked shot $\mathbf{s}_m$, respectively.

3.5 Fine-tuning for Scene Boundary Detection

Recall that we formulate the video scene segmentation as a binary classification task to identify contextual transition across the scene. Different from the pre-training stage, given an input sequence of shots $\mathbf{S}_n$, we employ a scene boundary detection head h_{SBD} to infer a prediction from the contextualized representation ($\mathbf{c}_n$) for the center shot $\mathbf{s}_n$. Following Chen et al. (2021), we freeze the parameters of the shot encoder and then train only the contextual relation network and the scene boundary detection head using a binary cross-entropy loss with the ground truth label y_n as follows:

$$\mathcal{L}_{\text{finetune}} = -y_n \log(h_{\text{SBD}}(\mathbf{c}_n)) + (1 - y_n)\log(1 - h_{\text{sbd}}(\mathbf{c}_n)). \quad (10)$$

Note that individual shots are decided to be a scene boundary when its prediction score is higher than a pre-defined threshold (set to 0.5).

Table 1: Comparison with other algorithms. † and ‡ denote that the numbers are copied from (Rao et al., 2020) and (Huang et al., 2020), respectively. * indicates the methods exploiting additional modalities or semantics (*e.g.*, audio, place, cast). The best numbers are highlighted in bold.

Method	AP (↑)	mIoU (↑)	AUC-ROC (↑)	F1 (↑)
Supervised Learning				
Siamese (Baraldi et al., 2015)‡	35.80	39.60	-	-
MS-LSTM (Huang et al., 2020)‡*	46.50	46.20	-	-
LGSS (Rao et al., 2020)†*	47.10	48.80	-	-
Unsupervised Learning				
GraphCut (Rasheed & Shah, 2005)†	14.10	29.70		-
SCSA (Chasanis et al., 2008)†	14.70	30.50	-	-
DP (Han & Wu, 2011)†	15.50	32.00	-	-
Story Graph (Tapaswi et al., 2014)†	25.10	35.70	-	-
Grouping (Rotman et al., 2017)‡*	33.60	37.20		-
BaSSL w/o fine-tuning (10 epochs)	31.55	39.36	71.67	32.55
Self-supervised Learning				
ShotCoL (Chen et al., 2021)	53.40	-	-	-
BaSSL (10 epochs)	56.26 $\pm$0.04	49.50 $\pm$0.11	90.27 $\pm$0.02	45.70 $\pm$0.24
BaSSL (40 epochs)	**57.40 $\pm$0.08**	**50.69 $\pm$0.45**	**90.54 $\pm$0.03**	**47.02 $\pm$0.87**

4 Experiment

4.1 Experimental Settings

Dataset We evaluate our proposed method on the MovieNet-SSeg dataset (Huang et al., 2020) that is a sub-dataset of MovieNet, containing 1,100 movies with 1.6M shots. Note that only 318 out of 1,100 movies have scene boundary annotations, which are divided into 190, 64, and 64 movies for training, validation, and test split, respectively. Following Chen et al. (2021), we use the entire 1,100 movies with no ground truth labels for the pre-training and fine-tune the model on the training split. The performance is measured on the test split.

Metric Following Huang et al. (2020), we compare algorithms using Average Precision (AP) and mIoU that measures the averaged intersection over union (IoU) between predicted scene segments and their closest ground truth scene segments. Also, we adopt F1 score and AUC-ROC as additional evaluation metrics. Note that contrary to the previous works (Rao et al., 2020; Chen et al., 2021) that report recall, we use F1 score to consider for balanced comparison between precision and recall. In addition, we report Meta-Sum metric inspired by the works (Chen et al., 2020b; Li et al., 2021) for easy and straightforward comparison of algorithms.

Implementation details We employ ResNet-50 (He et al., 2016) and Transformer (Vaswani et al., 2017) networks as the shot encoder and the contextual relation network, respectively. For both pre-training and fine-tuning stages, we cross-validate the number of neighbor shots among $K = \{4, 8, 12, 16\}$ and $K = 8$ is selected due to its good performance and computational efficiency. In all experiments, given a pre-trained model, we fine-tune the model 5 times with different random seeds and report their average score and standard deviation. Note that more details are presented in supplementary material.

4.2 Comparison with State-of-the-art Methods

We compare our method, BaSSL, with 1) supervised methods including Siamese (Baraldi et al., 2015), MS-LSTM (Huang et al., 2020) and LGSS (Rao et al., 2020), 2) unsupervised methods including GraphCut (Rasheed & Shah, 2005), SCSA (Chasanis et al., 2008), DP (Han & Wu, 2011), StoryGraph (Tapaswi et al., 2014) and Grouping (Rotman et al., 2017), and 3) self-supervised methods including ShotCoL (Chen et al., 2021). Without fine-tuning on the downstream task, BaSSL can be seen as an unsupervised model in that it is trained to predict the pseudo-boundary by the PP task. Table 1 summarizes comparison against state-of-the-art methods. BaSSL without fine-tuning shows competitive or outperforming performance based only on basic visual features compared to competing unsupervised methods. Furthermore, fine-tuning BaSSL with ground-truth scene bound-

Table 2: Average precision (AP) comparison with pre-training baselines. Note that SimCLR (NN) corresponds to our reproduced ShotCoL using SimCLR as the constrastive learning scheme.

	Method	Pre-training f_{ENC}	Pre-training f_{CRN}	Transfer f_{ENC}	Transfer f_{CRN}	Architecture of f_{CRN} during fine-tuning: MLP	MS-LSTM	Transformer
Supervised pre-training using image dataset								
M1	ImageNet	✓		✓		43.12 ±0.14	45.10 ±0.55	47.13 ±1.04
M2	Places365	✓		✓		43.82 ±0.10	45.87 ±0.40	48.71 ±0.50
Shot-level pre-training								
M3	SimCLR (instance)	✓		✓		45.60 ±0.07	49.09 ±0.24	51.51 ±0.31
M4	SimCLR (temporal)	✓		✓		45.55 ±0.11	49.24 ±0.26	50.05 ±0.78
M5	SimCLR (NN)	✓		✓		45.99 ±0.13	50.73 ±0.19	51.17 ±0.69
Boundary-aware pre-training								
M6	BaSSL	✓	✓	✓		46.53 ±0.11	50.58 ±0.14	50.82 ±0.69
M7	BaSSL	✓	✓	✓	✓	-	-	56.26 ±0.04
M8	M5+M7	✓	✓	✓	✓	-	-	56.86 ±0.01

aries, AP is improved by 24.71%p and BaSSL outperforms all other algorithms. Finally, through longer pre-training (40 epochs), BaSSL surpasses the state-of-the-art method (*i.e.*, ShotCoL) by a large margin (4.00%p in AP).

4.3 Comparison with Pre-training Baselines

We perform extensive experiments to compare BaSSL with the other pre-training baselines that learn shot-level representation by f_{ENC}. In the experiments, we compare the following three types of pre-training approaches; The first group (M1-2) trains f_{ENC} using image-level supervision with object labels on ImageNet (Deng et al., 2009) or place labels on Places365 (Zhou et al., 2017). The second group (M3-5) trains f_{ENC} through shot-level contrastive learning (*i.e.*, SimCLR proposed by Chen et al. (2020a)) with different positive pair sampling strategies. Specifically, *Instance* (M3) takes an instance of the center shot with different augmentation, *Temporal* (M4) takes one randomly sampled neighbor shot as positive pair in local temporal window, and *Nearest Neighbor (NN)* (M5) takes the most visually similar shot among the neighbor shots as positive pair, which is also known as ShotCoL (Chen et al., 2021). The last group (M6-8) learns both f_{ENC} and f_{CRN} through boundary-aware pretext tasks proposed in this paper. Given pre-trained representations, we train a video scene segmentation model with three different types of f_{CRN} including MLP (Chen et al., 2021), MS-LSTM (Huang et al., 2020)[1] and Transformer. For fair comparison, all pre-training methods employ ResNet-50 (He et al., 2016) as the shot encoder f_{ENC} and we pre-train the models for 10 epochs.

In Table 2, we found the following observations. First, when transferring pre-trained shot representation, employing MS-LSTM and Transformer as f_{CRN} is more effective than using MLP, as they are favorably designed to capture contextual relation between shots (see M1-6). Second, BaSSL (M7) outperforms all competing baselines (M1-5) through learning contextual representation during pre-training. Also, it turns out that transferring the representation through f_{CRN} is important for the boundary detection task where it leads to a performance gain of 5.44%p in AP (see M6-7). Finally, learning shot-level and contextual representations is complementary to each other; that is, incorporating ShotCoL (M5) and our framework (M7) provides further improved performance (M8).

4.4 Ablation Studies

Impact of individual pretext tasks We first investigate the contribution of individual pretext tasks. In this experiment, we train models by varying the combinations of the pretext tasks. From Table 3, we can obtain following two observations. First, when training a model with a single pretext task (P1-4), the MSM task leads to the worst performance compared to the others. This result indicates that boundary-aware pretext tasks (*i.e.*, SSM, CGM and PP) to learn contextual relation between shots is indeed important for video scene segmentation. Second, the more pretext tasks we include during pre-training, the better the performance is, and the best performance is obtained when using all tasks (P15). This means all tasks are complementary to each other, contributing to performance gain.

[1] https://github.com/AnyiRao/SceneSeg/tree/master/lgss

Table 3: Ablation study on varying combinations of pretext tasks for pre-training. The best scores are highlighted in bold.

	Pretext Tasks				Evaluation Metric				
	SSM	CGM	PP	MSM	AP	mIoU	AUC-ROC	F1	Sum
P1	✓				42.57 ±0.29	40.12 ±0.50	84.11 ±0.15	30.83 ±0.79	197.63
P2		✓			36.76 ±0.02	40.59 ±0.18	82.06 ±0.04	30.94 ±0.32	190.35
P3			✓		36.55 ±0.04	39.58 ±0.05	81.36 ±0.03	29.96 ±0.04	187.45
P4				✓	13.33 ±0.23	29.80 ±0.39	64.65 ±0.98	18.68 ±0.39	126.45
P5	✓	✓			55.77 ±0.05	48.19 ±0.21	90.19 ±0.03	43.17 ±0.39	237.32
P6	✓		✓		56.04 ±0.08	49.00 ±0.16	90.13 ±0.02	44.74 ±0.29	239.91
P7		✓	✓		38.09 ±0.03	41.25 ±0.10	82.85 ±0.01	32.24 ±0.24	195.43
P8	✓			✓	54.39 ±0.07	47.54 ±0.18	89.72 ±0.03	42.48 ±0.22	234.13
P9		✓		✓	39.49 ±0.04	41.71 ±0.12	83.27 ±0.02	32.85 ±0.20	197.32
P10			✓	✓	38.53 ±0.07	40.85 ±0.15	82.78 ±0.04	31.47 ±0.16	193.63
P11		✓	✓	✓	41.02 ±0.07	40.89 ±0.10	83.79 ±0.02	31.53 ±0.18	197.23
P12	✓		✓	✓	56.10 ±0.08	49.10 ±0.17	90.09 ±0.03	45.42 ±0.30	240.71
P13	✓	✓		✓	56.20 ±0.06	48.00 ±0.17	90.13 ±0.01	43.24 ±0.27	237.57
P14	✓	✓	✓		**56.26 ±0.02**	48.42 ±0.33	90.25 ±0.01	43.98 ±0.58	238.91
P15	✓	✓	✓	✓	**56.26 ±0.04**	**49.50 ±0.11**	**90.27 ±0.02**	**45.70 ±0.24**	**241.73**

Table 4: Ablations to check the impact of pseudo-boundary discovery strategies, the number of neighboring shots and longer pre-training. The best scores are in bold.

Pseudo-boundary	AP
Random	46.64 ±0.37
Fixed	49.53 ±0.32
Synthesized	54.61 ±0.03
DTW (ours)	**56.26 ±0.04**

(a) Performance comparison depending on the pseudo-boundary discovery methods.

# Neighbors	AP
4	55.98 ±0.10
8	56.26 ±0.04
12	**56.29 ±0.03**
16	55.31 ±0.04

(b) Performance comparison when varying the number of neighbor shots.

Epochs	AP
10	56.26 ±0.04
20	56.74 ±0.04
30	56.74 ±0.07
40	**57.40 ±0.08**
50	57.15 ±0.08

(c) Performance comparison with respect to the number of pre-training epochs.

Psuedo-boundary discovery method To check the effectiveness of DTW-based pseudo-boundary discovery, we train three models with different pseudo-boundary decision strategies—1) *Random* defining one randomly sampled shot in the input sequence as pseudo-boundary, 2) *Fixed* always taking the center shot as pseudo-boundary, and 3) *Synthesized*, inspired by Xu et al. (2020), synthesizing the input sequence by concatenating two sub-sequences sampled from different movies and using the last shot of the first sub-sequence as a pseudo-boundary. Table 4(a) summarizes the results. *Random* and *Fixed* pseudo-boundaries hinder the learning and degenerate the boundary detection performance. It is notable that BaSSL with *Synthesized* pseudo-boundaries also outperforms the pre-training baselines in Table 2, which shows the effectiveness of our framework. Finally, adopting DTW to find pseudo-boundaries achieves the best performance.

Hyperparameters We analyze the impact of two key hyperparameters: 1) the number of neighbor shots K and 2) pre-training epochs. Table 4(b) shows that we achieve higher performance with more neighbor shots, saturating around $K = 12$. Table 4(c) shows the impact of longer pre-training. We find that performance increases until certain numbers (*i.e.*, 40 epochs) and decrease afterward. We conjecture that this is partly due to overfitting to noise from incorrect pseudo-boundaries.

5 Conclusion

We present BaSSL, a novel self-supervised framework for video scene segmentation, especially designed to learn contextual relationship between shots. Through the pseudo-boundary discovery, we can define and conduct boundary-aware pretext tasks that encourage the model to learn the contextual relational representation and a capability of capturing transitional moments. Comprehensive experiments demonstrate the effectiveness of our framework and we achieve outstanding performance in the MovieNet-SSeg dataset.

Ethics statement In this paper, we introduce a novel self-supervised learning algorithm especially for video scene segmentation. The video scene segmentation technique could be applied to a wide range of applications, including preview generation for fast contents discovery, trailer generation, and minimally disruptive video-ads insertion. In addition, our proposed boundary-aware self-supervised learning algorithm can be applied to learn contextual representation of untrimmed, long videos. This would drive future research in videos to move towards leveraging the extremely large number of raw videos on the web to learn video representations. However, as a negative effect of this, it would consume large amount of computational resources. Also, as any machine learning algorithm is highly likely to be biased for training data, the learned representation would be biased and lead to an issue related to discrimination because the training data for self-supervised learning is often not validated by humans.

Reproducibility statement To make our algorithm reproducible, we present implementation details in Section 4.1 of the main paper and in Section B of the supplementary material; we also provide a pseudocode for DTW-based pseudo-boundary discovery method. In addition, we will release the code and parameters of trained models.

REFERENCES

Unaiza Ahsan, Chen Sun, and Irfan Essa. DiscrimNet: Semi-Supervised Action Recognition from Videos using Generative Adversarial Networks. *arXiv preprint arXiv:1801.07230*, 2018.

Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A Deep Siamese Network for Scene Detection in Broadcast Videos. In *Proceedings of the 23rd ACM international conference on Multimedia*, 2015.

Donald J Berndt and James Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *KDD workshop*, 1994.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3TW: Discriminative Differentiable Dynamic Time Warping for Weakly Supervised Action Alignment and Segmentation. In *CVPR*, 2019.

Vasileios T Chasanis, Aristidis C Likas, and Nikolaos P Galatsanos. Scene Detection in Videos using Shot Clustering and Sequence Alignment. *IEEE transactions on multimedia*, 11(1):89–100, 2008.

Shixing Chen, Xiaohan Nie, David Fan, Dongqing Zhang, Vimal Bhat, and Raffay Hamid. Shot Contrastive Self-Supervised Learning for Scene Boundary Detection. In *CVPR*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, 2020a.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV*, 2020b.

Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. TCLR: Temporal Contrastive Learning for Video Representation. *arXiv preprint arXiv:2101.07974*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-scale Hierarchical Image Database. In *CVPR*, 2009.

Yazan Abu Farha and Jurgen Gall. MS-TCN: Multi-Stage Temporal Convolutional Network for Action Segmentation. In *CVPR*, 2019.

Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A Large-Scale Study on Unsupervised Spatiotemporal Representation Learning. In *CVPR*, 2021.

Daniel Fried, Jean-Baptiste Alayrac, Phil Blunsom, Chris Dyer, Stephen Clark, and Aida Nematzadeh. Learning to Segment Actions from Observation and Narration. *arXiv preprint arXiv:2005.03684*, 2020.

Bo Han and Weiguo Wu. Video Scene Segmentation using A Novel Boundary Evaluation Criterion and Dynamic Programming. In *IEEE International conference on multimedia and expo*, 2011.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*, 2020.

Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. MovieNet: A Holistic Dataset for Movie Understanding. In *ECCV*, 2020.

Longlong Jing and Yingli Tian. Self-Supervised Spatiotemporal Feature Learning by Video Geometric Transformations. *arXiv preprint arXiv:1811.11387*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

Hilde Kuehne, Alexander Richard, and Juergen Gall. A hybrid RNN-HMM Approach for Weakly Supervised Temporal Action Segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):765–779, 2018.

Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised Learning of Action Classes with Continuous Temporal Embedding. In *CVPR*, 2019.

Sateesh Kumar, Sanjay Haresh, Awais Ahmed, Andrey Konin, M Zeeshan Zia, and Quoc-Huy Tran. Unsupervised Activity Segmentation by Joint Representation Learning and Online Clustering. *arXiv preprint arXiv:2105.13353*, 2021.

Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental Spatiotemporal CNNs for Fine-Grained Action Segmentation. In *ECCV*, 2016.

Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised Representation Learning by Sorting Sequences. In *ICCV*, 2017.

Jun Li and Sinisa Todorovic. Set-Constrained Viterbi for Set-Supervised Action Segmentation. In *CVPR*, 2020.

Jun Li and Sinisa Todorovic. Action Shuffle Alternating Learning for Unsupervised Action Segmentation. In *CVPR*, 2021.

Jun Li, Peng Lei, and Sinisa Todorovic. Weakly Supervised Energy-based Learning for Action Segmentation. In *ICCV*, 2019.

Linjie Li, Jie Lei, Zhe Gan, Licheng Yu, Yen-Chun Chen, Rohit Pillai, Yu Cheng, Luowei Zhou, Xin Eric Wang, William Yang Wang, et al. VALUE: A Multi-Task Benchmark for Video-and-Language Understanding Evaluation. In *NeurIPS*, 2021.

Chao Liang, Yifan Zhang, Jian Cheng, Changsheng Xu, and Hanqing Lu. A Novel Role-Based Movie Scene Segmentation Method. In *Pacific-Rim Conference on Multimedia*, pp. 917–922. Springer, 2009.

Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. In *ECCV*, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018.

Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal Contrastive Video Representation Learning. In *CVPR*, 2021.

Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. A Local-to-Global Approach to Multi-modal Movie Scene Segmentation. In *CVPR*, 2020.

Zeeshan Rasheed and Mubarak Shah. Scene Detection in Hollywood Movies and TV Shows. In *CVPR*, 2003.

Zeeshan Rasheed and Mubarak Shah. Detection and Representation of Scenes in Videos. *IEEE transactions on Multimedia*, 7(6):1097–1105, 2005.

Byungseok Roh, Wuhyun Shin, Ildoo Kim, and Sungwoong Kim. Spatially Consistent Representation Learning. In *CVPR*, 2021.

Daniel Rotman, Dror Porat, and Gal Ashour. Robust and efficient video scene detection using optimal sequential grouping. In *2016 IEEE international symposium on multimedia (ISM)*, 2016.

Daniel Rotman, Dror Porat, and Gal Ashour. Optimal Sequential Grouping for Robust Video Scene Detection using Multiple Modalities. *International Journal of Semantic Computing*, 11(02):193–208, 2017.

Yong Rui, Thomas S Huang, and Sharad Mehrotra. Exploring Video Structure beyond The Shots. In *Proceedings. IEEE International Conference on Multimedia Computing and Systems (Cat. No. 98TB100241)*, 1998.

Yuhan Shen, Lu Wang, and Ehsan Elhamifar. Learning To Segment Actions From Visual and Language Instructions via Differentiable Weak Sequence Alignment. In *CVPR*, 2021.

Panagiotis Sidiropoulos, Vasileios Mezaris, Ioannis Kompatsiaris, Hugo Meinedo, Miguel Bugalho, and Isabel Trancoso. Temporal Video Segmentation to Scenes using High-level Audiovisual Features. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(8):1163–1177, 2011.

Yaser Souri, Mohsen Fayyaz, Luca Minciullo, Gianpiero Francesca, and Juergen Gall. Fast Weakly Supervised Action Segmentation using Mutual Consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised Learning of Video Representations using LSTMs. In *ICML*, 2015.

Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning Video Representations using Contrastive Bidirectional Transformer. *arXiv preprint arXiv:1906.05743*, 2019a.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A Joint Model for Video and Language Representation Learning. In *ICCV*, 2019b.

Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. StoryGraphs: Visualizing Character Interactions as a Timeline. In *CVPR*, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NIPS*, 2017.

Rosaura G VidalMata, Walter J Scheirer, Anna Kukleva, David Cox, and Hilde Kuehne. Joint Visual-Temporal Embedding for Unsupervised Learning of Actions in Untrimmed Sequences. In *WACV*, 2021.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating Videos with Scene Dynamics. *NIPS*, 2016.

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking Emerges by Colorizing Videos. In *ECCV*, 2018.

Zhe Wang, Hao Chen, Xinyu Li, Chunhui Liu, Yuanjun Xiong, Joseph Tighe, and Charless Fowlkes. Unsupervised Action Segmentation with Self-supervised Feature Learning and Co-occurrence Parsing. *arXiv preprint arXiv:2105.14158*, 2021.

Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-Supervised Spatiotemporal Learning via Video Clip Order Prediction. In *CVPR*, 2019.

Mengmeng Xu, Juan-Manuel Pérez-Rúa, Victor Escorcia, Brais Martinez, Xiatian Zhu, Li Zhang, Bernard Ghanem, and Tao Xiang. Boundary-sensitive Pre-training for Temporal Localization in Videos. *arXiv preprint arXiv:2011.10830*, 2020.

Yang You, Igor Gitman, and Boris Ginsburg. Large Batch Training of Convolutional Networks. *arXiv preprint arXiv:1708.03888*, 2017.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-Task Weakly Supervised Learning from Instructional Videos. In *CVPR*, 2019.

A Dataset Statistics

Table 5 shows the data statistics of different video scene segmentation datasets. We found the limited number of datasets that provide the scene boundary annotations and, as far as we know, the MovieNet-SSeg (Huang et al., 2020) is the largest-scale video scene segmentation dataset.

Table 5: Comparison of existing video scene segmentation datasets. Note that we brought the table from (Rao et al., 2020) with an update on the MovieNet-SSeg dataset.

Dataset	#Video	#Scene	#Shot	Time (h)	Source
BBC (Baraldi et al., 2015)	11	670	4.9K	9	Documentary
OVSD (Rotman et al., 2016)	21	300	10K	10	MiniFilm
MovieNet-SSeg (Huang et al., 2020)	318	42K	500K	-	Movies
MovieNet (Huang et al., 2020)	1,100	-	1.6M	-	Movies

B Additional Implementation Details

Additional details of the shot encoder (*i.e.*, ResNet-50) and the contextual relation network (*i.e.*, Transformer) are as follows. For Transformer, the dimensions are set to $(L = 2, H = 768, A = 8)$ where L, H and A mean the number of stacked transformer blocks, the dimension of hidden activation and the number of attention heads, respectively. We apply the Dropout technique (Srivastava et al., 2014) on hidden states and attention weights with a probability of 10% and use GELU (Hendrycks & Gimpel, 2016) as an activation function. For the shot encoder, each shot is given by three key-frames (*i.e.*, $N_k = 3$) and a shot encoding $\mathbf{e}$ is computed by the averaged feature after inferring individual three key-frames by ResNet-50; note that, to speed up the training, we use randomly sample one key-frame out of three during the pre-training.

For data augmentation of key-frames in a shot, we adopt PyTorch's torchvision package. Given a sequence of shots, we apply random crop (with resize), random flip, random color jitter and random Gaussian blur. In detail, firstly, the cropping is performed with a random size (*i.e.*, scales between [0.14, 1.0] of the original size) and a random aspect ratio (between 3/4 to 4/3), and then the cropped one is resized to (224,224). Secondly, we apply a random horizontal flip with a probability of 50%. Thirdly, as a color augmentation, we perform a random color jitter (with a probability of 80%) and a random color dropping to gray scale (with a probability of 20%). The color strength parameters for jittering are set to $\{\text{brightness}, \text{contrast}, \text{saturation}, \text{hue}\} = (0.2, 0.2, 0.2, 0.05)$. Finally, Gaussian blur is applied with a probability of 50% where a standard-deviation of spatial kernel is set to [0.1, 2.0]. Note that the same augmentations are applied to all key-frames in the input sequence $\mathbf{S}_n$ of shots while different color jittering is applied on individual shots. Also, for $\mathbf{S}_n^{\text{slow}}$, we perform a different augmentation compared to that applied on $\mathbf{S}_n$.

During the pre-training stage, the model parameters are randomly initialized and then trained using the proposed pretext tasks. We use LARS (You et al., 2017) to learn the model (except for parameters of bias and Batch-Normalization) with a mini-batch of 256 shot sequences, a base learning rate of 0.3, momentum of 0.9, weight decay of 10^{-6} and trust coefficient of 0.001. 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. The temperature τ in Eq. (5) is set to 0.1.

In the fine-tuning stage, we initialize the parameters of the shot encoder and the contextual relation network by that of the pre-trained ones, however, we freeze the parameters of the shot encoder following Chen et al. (2021). We fine-tune the contextual relation network and the scene boundary detection head for 20 epochs using Adam (Kingma & Ba, 2015) with a learning rate of 10^{-5} and a mini-batch of 1024 training examples. The learning rate is decayed with a cosine schedule without a warm-up stage.

C Comparison with Shot-level Self-supervised Learning

As mentioned in the main paper, our approach is distinguishable from the shot-level pre-training approach (Chen et al., 2021; Qian et al., 2021) in that the objectives used in our approach (BaSSL)

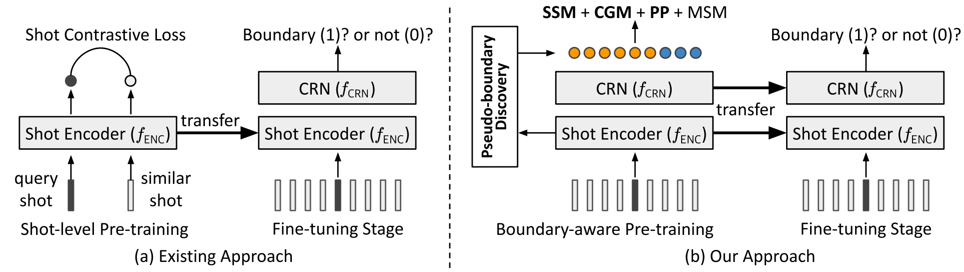

Figure 5: Comparison between existing approaches and ours for video scene segmentation. The existing approach focuses only on learning shot-level representation given by shot encoder (f_{ENC}). In contrast, our boundary-aware pre-training method focuses on learning contextual representation by taking neighbor shots into account. Thus, our method can learn both the shot encoder (f_{ENC}) and the contextual relation network (CRN; f_{CRN}) and transfer their parameters during the fine-tuning stage.

Table 6: Comparison between the shot-level pre-training and the proposed pre-training approaches.

Check List	Shot-level Pre-training	Boundary-aware Pre-training
Network architecture	f_{ENC}	f_{ENC} + f_{CRN}
Training input	a pair of shots (#shots: 2)	a sequence of shots (#shots: 2K+1)
Weights transferable for f_{ENC}?	yes	yes
Weights transferable for f_{CRN}?	no	yes
Positive pair in contrastive learning	shot-shot	shot-scene

is to learn contextual representations by taking neighbor shots into account. Figure 5 provides a clear summary of comparison between shot-level pre-training and our BaSSL. Firstly, shot-level pre-training takes a pair of two shots as an input while BaSSL takes a sequence of shots. Secondly, shot-level pre-training aims to train shot encoder (f_{ENC}) only, while BaSSL trains both the shot encoder and the contextual relation network (f_{ENC} and f_{CRN}). In contrast to the shot-level pre-training that requires to train f_{CRN} from scratch during the fine-tuning stage, BaSSL benefits from weight transfer by pre-training the parameters of f_{CRN} with large-scale in-domain data in advance. Note that the results (M6-7) in Table 2 show that the weight transfer of f_{CRN} is important to improve the video scene segmentation performance. Finally, the contrastive learning objective in shot-level pre-training locates the representations of two shots (query and positive) to be close to each other, whereas Shot-Scene Matching objective in our approach performs the same task but with a shot (query) and its associated scene (positive; a sequence of shots). The Table 6 summarizes the aforementioned comparisons.

D Algorithm for Pseudo-boundary Discovery

In this section, we describe the details of pseudo-boundary discovery method applying DTW on $\mathbf{S}_n$ and $\mathbf{S}_n^{\text{slow}}$. In practice, $\mathbf{S}_n$ is given as a mini-batch resulting in a tensor with a shape of (B, S, N_k, C, H, W) where individuals mean the batch size, the number of shots in $\mathbf{S}_n$ (*i.e.*, $2K+1$), the number of key-frames in a shot, channels, frame height and frame width, respectively. Then, we obtain $\mathbf{S}_n^{\text{slow}} \in \mathbb{R}^{B\times 2\times N_k\times C\times H\times W}$ that is composed of the first and last shots in $\mathbf{S}_n$. We apply two different augmentation functions into key-frames in $\mathbf{S}_n$ and $\mathbf{S}_n^{\text{slow}}$, respectively. Next, we compute encoded representation of $\mathbf{S}_n$ and $\mathbf{S}_n^{\text{slow}}$ using f_{ENC}. Note that during the pre-training stage, we randomly sample one key-frame among N_k candidates in a shot and then reshape the input tensor as (B*S, C, H, W) or (B*2, C, H, W) to be forwarded by the shot encoder f_{ENC}; thus the tensor shape of the encoded shot representation is given by (B, S, D_e) or (B, 2, D_e) after apply reshaping, where D_e means the dimension of encoded feature. Finally, given two sequences of encoded representation for $\mathbf{S}_n$ and $\mathbf{S}_n^{\text{slow}}$, DTW provides two sub-sequences $\mathbf{S}_n^{\text{left}}$ and $\mathbf{S}_n^{\text{right}}$ and a pseudo boundary shot $\mathbf{s}_{n+b^*}$. The algorithm 1 illustrates the details. In addition, to demonstrate the simplicity of the alignment computation using DTW, we include the PyTorch code in Listing 1. The implementation of DTW can be done in 5 lines of python code using *tslearn* package.

```python
from tslearn import metrics
import numpy as np
def compute_dtw_path(self, seq_1, seq_2):
    """
    Input:
        seq_1: sparse shots embedding, shape = torch.Size([2, dim])
        seq_2: dense shots embedding, shape = torch.Size([N, dim]), N > 2
    Output:
        dtw_path: output of DTW algorithm, shape = torch.Size([N, dim])
    """
    cost = (1-torch.bmm(seq_1, seq_2.transpose(1, 2))).numpy()
    dtw_path = []
    for bsz in range(cost.shape[0]):
        _path, _ = metrics.dtw_path_from_metric(cost[bsz], metric="precomputed")
        dtw_path.append(np.asarray(_path)) # torch.Size([N, dim])
    return dtw_path
```

Listing 1: PyTorch code for alignment computation using DTW given two sequences. The *tslearn* package is used for DTW path calculation.

Algorithm 1 DTW-based pseudo-boundary discovery

1: **Input**: Shot encoder f_{enc}, contextual relation network f_{CRN}, and an input shot sequence $\mathbf{S}_n = \{\mathbf{s}_{n-K}, ..., \mathbf{s}_n, ..., \mathbf{s}_{n+K}\}$ centered at n^{th} shot $\mathbf{s}_n$ with neighbor size K, two image augmentation functions $\lambda^1_{\text{aug}}, \lambda^2_{\text{aug}}$.
2: $(\mathbf{E}_n, \mathbf{E}^{\text{slow}}_n) \leftarrow$ ([], [])
3: **for** $i = n - K$ **to** $n + K$ **do**
4: $\quad \mathbf{e}_i \leftarrow f_{\text{ENC}}(\lambda^1_{\text{aug}}(\mathbf{s}_i))$ // extract shot-level representations for all shots
5: $\quad \mathbf{E}_n \leftarrow \{\mathbf{E}_n; \mathbf{e}_i\}$ // append
6: **end for**
7: **for** i **in** $\{n - K, n + K\}$ **do**
8: $\quad \mathbf{e}_i \leftarrow f_{\text{ENC}}(\lambda^2_{\text{aug}}(\mathbf{s}_i))$ // extract shot-level representations for slow sequence
9: $\quad \mathbf{E}^{\text{slow}}_n \leftarrow \{\mathbf{E}^{\text{slow}}_n; \mathbf{e}_i\}$ // append
10: **end for**
11: $\mathbf{S}^{\text{left}}_n, \mathbf{S}^{\text{right}}_n, b^* \leftarrow \text{DTW}(\mathbf{E}_n, \mathbf{E}^{\text{slow}}_n)$ // apply dynamic time warping
12: **Output**: Two continuous non-overlapping sub-sequences $\mathbf{S}^{\text{left}}_n$ and $\mathbf{S}^{\text{right}}_n$ and a pseudo boundary shot $\mathbf{s}_{n+b^*}$.

E RESULTS ON ADDITIONAL DATASETS

We further compare BaSSL with shot-level pre-training baselines on additional two datasets—BBC and OVSD. Note that the train and test splits are not available and the dataset size is extremely limited (11 and 21 videos in BBC and OVSD, respectively); in addition, 2 out of 21 videos in OVSD is not available. Thus, we infer predictions using models trained on MovieNet-SSeg without fine-tuning on BBC and OVSD. The results are summarized in Table 7. The result shows the superiority of our method compared to shot-level pre-training baselines.

Table 7: Comparison between our method and shot-level pre-training baselines on BBC and OVSD datasets. The numbers mean AP.

Model	SimCLR (instance)	SimCLR (temporal)	SimCLR (NN)	BaSSL
BBC	32.34	34.18	32.92	**39.98**
OVSD	25.45	24.92	25.02	**28.68**

Table 8: Scene clustering quality measured by normalized mutual information (NMI) metric.

Model	Scene Length			$\Delta\downarrow$ (Short $\rightarrow$ Long)
	Short (N_c=8)	Medium (N_c=16)	Long (N_c=32)	
ImageNet	67.50	61.60	56.25	-16.67%
SimCLR (temporal)	82.40	81.65	78.99	-4.14%
SimCLR (NN)	83.54	83.17	81.25	-2.75%
BaSSL (ours)	**86.22**	**86.72**	**85.63**	**-0.68%**

Table 9: Ablation study on the combination of boundary-aware pretext tasks measured by NMI.

Pretext Tasks	NMI	Gain (Δ%)
SSM	85.48	0.00%
SSM+MSM	85.64	+0.19%
SSM+MSM+CGM	85.93	+0.33%
SSM+MSM+CGM+PP	86.71	+0.91%

F MEASURING REPRESENTATION QUALITY AT PRE-TRAINING STAGE

The normalized mutual information (NMI) is a metric for clustering algorithms (*e.g.*, K-Means), which measures the clustering quality. Since clustering with good representations forms clear boundaries between different classes, NMI can be considered as a proxy to measure the quality of our pre-trained models. Specifically, we randomly sample 100 scenes from the test split of MovieNet-SSeg while we vary the length of scenes $N_c \in \{8, 16, 32\}$. Then, we perform K-Means clustering on $N_c \times 100$ shot representations extracted by the pre-trained model with the number of classes K=100. We intend to form a single cluster for each scene in this formulation, assuming that high-quality representation for movie scene segmentation would locate the shot features within the same scene close to each other. Considering the randomness in the K-Means clustering and scene sampling, we report the averaged score from five trials.

In Table 8, we compare the NMI score between different pre-trained models; SimCLR (NN) is our SimCLR version implementation of ShotCoL. The result shows that BaSSL outperforms the shot-level pre-training baselines and the model pre-trained using ImageNet dataset. With respect to different scene lengths (N_c; the number of shots included in a single scene), we found our BaSSL is more robust than the other baselines. Since the visual diversity across the shots increases as the scenes become longer (N_c=8 $\rightarrow$ 32), it is natural that the NMI score for each baseline is degraded. However, it is remarkable that, by increasing the number of shots from 8 to 32, the performance of BaSSL drops only -0.68% while the other baselines suffer from severe degradation. This demonstrates the effectiveness of BaSSL in maximizing intra-scene similarity.

In addition, we perform ablation study of our algorithm by adding pretext tasks one by one, and measure the corresponding NMI scores. The result in Table 9 shows that better NMI score is achieved as more pretext tasks are combined together. This tendency is also observed in our ablation in Table 3, which indicates the NMI score of pre-trained models is highly correlated with the final performance after the fine-tuning.

G QUALITATIVE ANALYSIS

Visualization of similarities between consecutive shots To qualitatively check the effect of individual pretext tasks, we visualize the matrix of cosine similarity between shot representations from the randomly sampled 16 consecutive shots in Figure 6. The shot representations are computed by models without the fine-tuning in order to solely focus on the behavior of each objective at the pretraining stage. When the MSM is used only, approximately three clusterings are shown, but similarity around boundaries is smoothed. Next, when we add PP, dissimilarities around the boundaries are to be sharpened. Then, with additional CGM, the clusters are more clearly obtained. Finally, adding SSM makes the similarity of shots within the same cluster higher (i.e., more yellow ones).

Pseudo-boundaries We compare the quality of discovered pseudo-boundaries with the ground truth scene boundaries in Figure 7. In most cases, we observe the pseudo-boundaries identified by

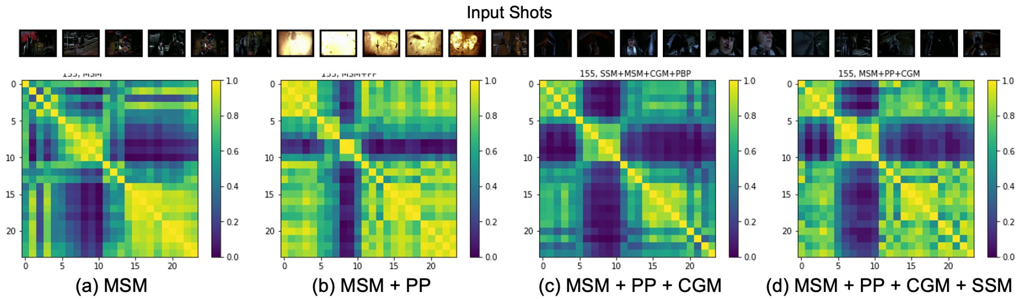

Figure 6: Visualization of similarity (below) between shot representations in randomly sampled consecutive shots (above). We observe that the shot representations are clearly clustered as adding pretext tasks one by one.

the DTW algorithm are successfully located in close distance with the ground truth ones. This result validates our idea considering the problem of discovering pseudo-boundary as a temporal alignment problem between two sequences with different frequencies ($\mathbf{S}_n$ and $\mathbf{S}_n^{\text{slow}}$). At the same time, we illustrate the failure cases. Although discovered pseudo-boundary does not match the ground truth in this case, we figure the determined boundary is not always arbitrary. For example, the mismatch is often caused by the noise existing in the ground truth (see the first row in the failure cases). On the other hand, in case all shots are visually similar (see the third row in the failure cases), the DTW solely relying on the visual modality fails to find the correct boundary.

Predicted scene boundaries The figure 8 illustrates the scene boundary predictions of different models. Comparing with the baselines, we observe that our approach, BaSSL, shows qualitatively better performance for video scene segmentation. On the other hand, we observe the over-segmentation issue in many cases using any compared methods (including ours). Our finding implies that achieving the highest recall only does not guarantee the highest performance in practice. We reckon that further studies on this over-segmentation problem would be a highly important topic when it comes to real-world application.

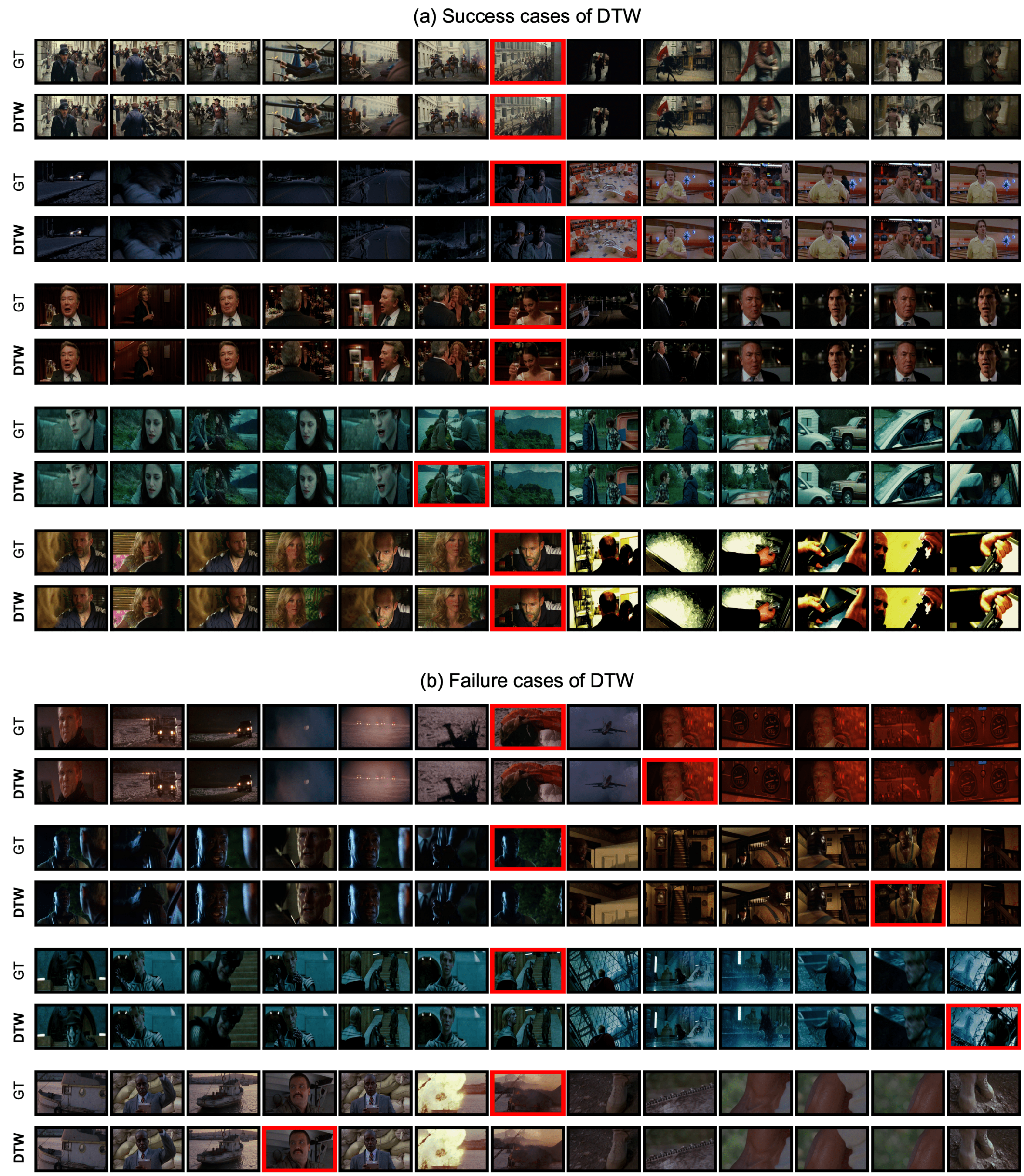

Figure 7: Comparison between the ground truth scene boundaries and the discovered pseudo-boundaries based on the DTW algorithm. The examples are sampled from the MovieNet-SSeg dataset. All boundary shots are highlighted in red.

Figure 8: Comparison of boundary detection results from three pre-training approaches: ImageNet pre-trained ResNet, ShotCoL, and BaSSL. The first row shows the reference that is composed of two adjacent scenes divided by the ground truth boundary. We visualize the shots that are assigned to the same scene segments with the same colored border.