Unsupervised Audio-Visual Lecture Segmentation
Supplementary Material

Darshan Singh S* Anchit Gupta* C. V. Jawahar Makarand Tapaswi
CVIT, IIIT Hyderabad
https://cvit.iiit.ac.in/research/projects/cvit-projects/avlectures

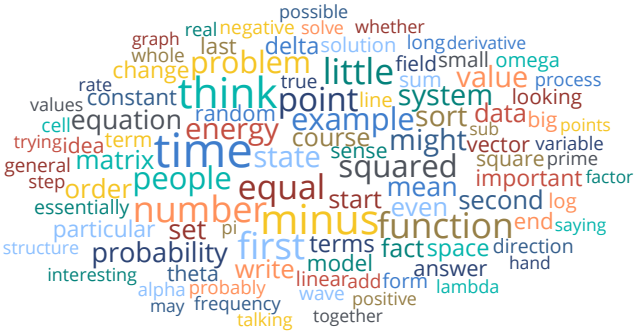

Figure 1: Wordcloud summarizing the distribution of frequently occurring words in AVLectures.

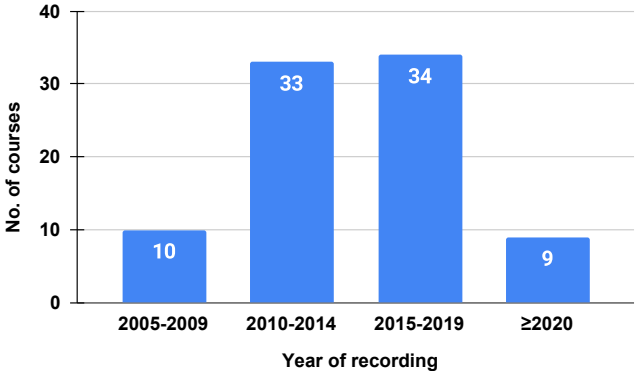

Figure 2: Courses from AVLectures that are recorded over the last 2 decades.

The supplementary material is structured as follows: In Sec. A we present details about the vocabulary of AVLectures. Next, we analyze a failure case example of segmentation in Sec. B. In Sec. C we provide details on segmenting the lectures manually. Further, we discuss some more ablation studies in Sec. D and training details in Sec. E. Next, we provide additional qualitative results for both the text-to-video retrieval as well as the lecture segmentation task in section Sec. F. Finally, we report segmentation scores for each of the 15 courses in Sec. G.

A. AVLectures Dataset: Additional Details

AVLectures has a vocabulary size of around 13,000 words with over 7.1M words in total. Fig. 1 shows the distribution of the most occurring words in the dataset. AVLectures is currently dominated by STEM courses, primarily Electrical Engineering & Computer Science, Physics, and Mathematics, which is evident from the word cloud in Fig. 1. In our dataset, we have a good mix of old and new courses, as seen in Fig. 2, with the majority being recorded in the last decade.

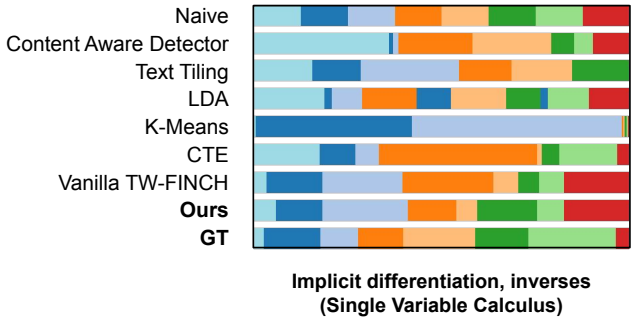

Figure 3: Example of segmentation where the predicted segments differ slightly from the ground truth segments.

B. Deep dive into the failure case

In this section, we provide more insights into a failure case example of segmentation.

Consider Fig. 3, in which the predicted segments from our model are slightly different from the ground truth segments. However, segmentation is a subjective task, and there can be more than one valid segmentation for some lectures. Our aim in audio-visual lecture segmentation is to temporally segment a lecture into several smaller segments,

such that each segment represents a unique concept/subtopic. Consider a case in which a single concept can be divided into two smaller concepts. In this case, two valid segmentations are possible: (i) the single concept considered as one complete segment or (ii) the two smaller concepts considered as two separate segments.

Now we compare the Ground Truth (GT) segmentation with the segmentation predicted by our model for the lecture *Implicit differentiation* of the *Single Variable Calculus* course[1]. We will refer to the i^{th} segment of Ground Truth as GT_i and that of the segment predicted by our model as $Pred_i$ for the rest of this section. Also, let $len(segment)$ represent the total duration (or the length) of the $segment$.

GT_1 and $Pred_1$ are about the *introduction to implicit differentiation*. However, $Pred_1$ is slightly longer as it includes the part where the *professor greets the late-coming students*. However, this non-lecture segment is a part of GT_2 and the rest of it is similar to the $Pred_2$, which covers the topic of *the rational exponent rule*. The ending boundary of GT_2 is approximately equal to that of $Pred_2$. Also,

$$len(GT_1) + len(GT_2) \approx len(Pred_1) + len(Pred_2)$$

GT_3 discusses the calculation of the *slope of the tangent to a circle using the direct method* and GT_4 *using the implicit method*. However, $Pred_3$ combines both the segments into one. The ending boundary of GT_4 is close to that of $Pred_3$. Also,

$$len(GT_3) + len(GT_4) \approx len(Pred_3)$$

Next, GT_5 is an example involving a *fourth-order equation*. In the case of predicted segmentation, the example is divided into two segments $Pred_4$ and $Pred_5$, which correspond to the two steps involved in solving it. This is an error made by our model as it breaks the two-step solution, however, it is nice to observe that the split is still at a meaningful location. The ending boundary of GT_5 is approximately equal to that of $Pred_5$. Also,

$$len(GT_5) \approx len(Pred_4) + len(Pred_5)$$

The last three segments of GT are about the *derivatives of inverse functions and a couple of examples*. Among the last three predicted segments, segment 6 and segment 7 are about the *derivatives of inverse functions* and the problem statement of the examples. The final predicted segment covers the solution to both the examples.

$$\begin{aligned} &len(GT_6) + len(GT_7) + len(GT_8) \\ &\approx len(Pred_6) + len(Pred_7) + len(Pred_8) \end{aligned}$$

Hence, even though the predicted segmentation is slightly different from the GT segmentation, it is still a valid segmentation.

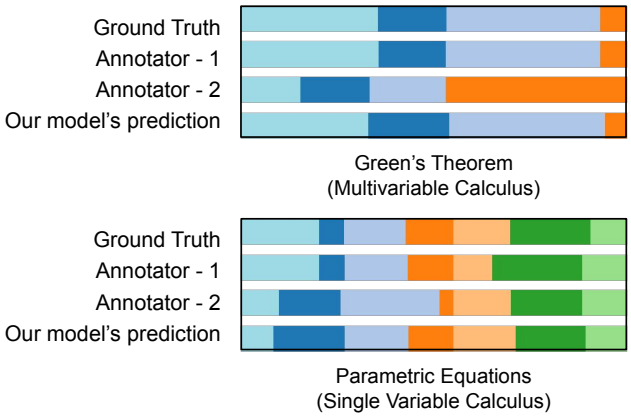

Figure 4: Segmentation of two lectures done manually by two annotators.

Lecture	Method	NMI	MOF	IOU	F1	BS@30
Green's Theorem	A-1	98.0	99.6	99.0	99.5	100.0
	A-2	76.1	75.2	55.8	63.1	66.7
	Ours	86.7	95.4	88.8	94.0	66.7
Parametric Equations	A-1	89.7	92.6	87.8	93.0	66.7
	A-2	81.4	77.5	63.1	74.2	50.0
	Ours	86.6	84.8	76.3	84.1	66.7

Table 1: Inter-Annotator segmentation scores. Here, A-1 stands for Annotator-1, A-2: Annotator-2, Ours: Our model's prediction.

C. Inter-annotator variation

To further analyze the subjective nature of the segmentation task, we asked two annotators to independently segment a few lectures to check the agreement with the corresponding ground truth segmentation and among themselves. Fig. 4 shows two such results. In the first example, Annotator-1 considered the topic and its example to be the same segment, whereas Annotator-2 split them into two separate segments. We can see that even though the Annotator-2's segmentation does not match with the Ground Truth, it is still a valid segmentation. Also, our model predicts segments that are closer to that of GT and Annotator-1. In the second example, our model predicts the first two segments in line with Annotator-2's segments while the rest of the segments are similar to that of GT and Annotator-1's segments. In Table 1, we provide quantitative results of segmentation done by each of the annotators, as well as the prediction from our model with respect to MIT OCW's ground truth.

Method	Partition	NMI ↑	MOF ↑	IOU ↑	F1 ↑	BS@30 ↑
Ours	2nd last	63.7	61.7	59.5	42.6	42.3
	3rd last	72.1	59.7	39.1	42.7	65.2
	GT	79.8	80.3	69.2	76.9	58.7
Naive	2nd last	58.6	58.9	54.1	40.5	27.0
	3rd last	66.9	51.2	33.8	39.3	38.9
	GT	71.8	75.5	62.7	74.0	32.5

Table 2: Allowing TW-FINCH to estimate the number of clusters.

D. Additional Ablation studies

1. What if the number of segments is unknown? It is not trivial to guess the ideal number of segments for the unseen lectures. In such cases, we let the TW-FINCH algorithm decide the appropriate number of clusters. TW-FINCH produces a hierarchy of partitions where the number of clusters reduces with successive partitions. We use the 2nd- and the 3rd-last partitions to estimate the number of segments automatically and report performance in Table 2. We also report scores for the Naïve baseline on the above partitions as well.

In addition to the usual metrics we also compute the L1 distance between the ground-truth number of clusters and the number of automatically estimated clusters for both the partitions. The L1 distance between the last and 2nd-last partition is 8.554 and that of 3rd-last is 4.614. The 3rd-last partition has a lower L1 score compared to the 2nd-last partition. This, along with the other metrics, indicates that the number of clusters generated by the 3rd-last partition is closer to the ground-truth.

Language Model	NMI ↑	MOF ↑	IOU ↑	F1 ↑	BS@30 ↑
Word2Vec	78.9	79.7	68.4	76.4	58.2
mpnet-v1	79.1	79.7	68.3	76.2	58.4
mpnet-v2	**79.8**	**80.3**	**69.2**	**76.9**	**58.7**

Table 3: Impact of different Language Models.

2. Using different language embedding models. In this study, we experiment with three different text embeddings,

1. word2vec: We first preprocess the transcripts by removing the most common stop words. Next, we extract the word embeddings from the GoogleNews pretrained word2vec model [3]. word2vec encodes each word into to a 300-dimensional vector.

2. `multi-qa-mpnet-base-dot-v1` (mpnet-v1 in Table 3): This is a sentence transformer BERT model that uses the pre-trained MPNet [4] model and is trained on 215M (question, answer) pairs from diverse sources. This model encodes the transcripts into a 768-dimensional vector.

3. `all-mpnet-base-v2` (mpnet-v2 in Table 3): This model uses the pre-trained MPNet [4] model and is fine-tuned on a 1B sentence pairs dataset using a contrastive learning objective: given a sentence from the sentence pairs, the model should predict which sentence from a randomly sampled other sentences was paired with it. This is the same model that was described in the Main paper Sec. 4.1.

The results of all three models are reported in Table 3. Although, the `all-mpnet-base-v2` model performs slightly better when compared to the other two text embedding models the scores are almost similar in all three variations. The results show that there is no significant impact on the type of text embeddings that are used to train the model.

Embed. dim.	NMI ↑	MOF ↑	IOU ↑	F1 ↑	BS@30 ↑
512	79.3	79.7	68.3	76.1	**59.7**
1024	79.3	80.3	68.9	76.7	59.0
2048	**79.8**	**80.4**	**69.4**	**77.1**	59.6
4096	**79.8**	80.3	69.2	76.9	58.7

Table 4: Impact of different embedding dimension.

3. How does the model's embedding dimension affect the performance of segmentation? We train the model with four different output embedding dimensions: 512, 1024, 2048 and 4096. It can be seen from Table 4 that the learned features are robust and independent of the feature dimension and therefore has little impact on the overall performance of the model on the segmentation task. Although the embedding dimensions 2048 and 4096 perform slightly better than the rest.

	Feature modality							
	visual	textual	learned	NMI ↑	MOF ↑	IOU ↑	F1 ↑	BS@30 ↑
1	✓	-	✗	53.1	58.6	38.2	46.2	37.5
2	-	✓	✗	48.5	55.1	33.5	41.0	34.3
3	✓	✓	✗	53.1	58.9	38.6	46.5	37.9
4	✓	-	✓	**63.9**	**66.8**	**48.2**	**55.7**	**44.9**
5	-	✓	✓	49.2	56.4	35.0	42.4	33.7
6	✓	✓	✓	60.2	64.9	46.0	53.3	44.1

Table 5: Impact of different feature modalities on K-Means

4. Impact of different feature modalities on K-Means and CTE [1] We show the segmentation results for K-means and Continuous Temporal Embedding [1] (CTE) on the features extracted using the pipeline (Sec. 4.1 Main Paper) as well as on the learned embeddings from our joint text-video model. The scores are shown in Table 5 and 6.

[1] Implicit differentiation lecture video

	Feature modality			NMI ↑	MOF ↑	IOU ↑	F1 ↑	BS@30 ↑
	visual	textual	learned					
1	✓	-	✗	65.0	65.4	45.9	55.4	38.6
2	-	✓	✗	**67.2**	**68.1**	**49.6**	**59.4**	35.3
3	✓	✓	✗	66.3	66.5	47.4	57.0	39.8
4	✓	-	✓	67.1	67.2	48.2	57.6	41.0
5	-	✓	✓	64.7	65.7	45.4	54.8	35.6
6	✓	✓	✓	**67.2**	67.3	48.1	57.3	**41.5**

Table 6: Impact of different feature modalities on CTE

For K-Means, the learned visual embeddings (row 4) and the combination of learned visual and textual embeddings (row 6) outperforms all other variations by a good margin. The results highlight the importance of training lecture-aware representations using our joint text-video embedding model. For CTE, even though all the scores are relatively closer to each other, the one that uses text features (BERT embeddings) (row 2) and a combination of learned visual and textual embeddings (row 6) perform the best. Note that using a combination of learned visual and textual embeddings results in the highest boundary score, highlighting the importance of our learned representations in predicting better boundaries.

5. Deeper analysis on Naïve method performing well. As discussed in the paper, one reason why the Naïve method is effective is due to an inherent bias of the instructor spending almost equal amounts of time on different topics in certain lectures. For example, consider a lecture on Multivariate Calculus[2]. Here each of the segment is approximately 16 minutes, thus giving an upper-hand to the naive method. Upon further analysis, we observe that 73 of 350 lectures (nearly 20 % of CwS) have GT segment boundaries within 3 minutes to the boundaries suggested by the Naïve baseline. We perform an ablation study by varying the number of splits obtained by automatically clustering lectures with TW-FINCH. The results indicate that splitting lectures at the ground truth number of segments gives a better segmentation performance than splitting it in any other way, as seen in Table 2.

6. Boundary scores at various intervals. We also perform an ablation study by computing Boundary Scores at various values of K, and it's plot is shown in Fig. 5. Typically, the instructor spends at least 25-30 seconds (in answering student's questions, erasing the blackboard etc.) before switching to new a topic. This was the reason behind reporting the scores for BS@30 in the paper. As expected, all methods perform worse for lower values of K and as K approaches 15, the use of 10-15s clip sizes hurts performance.

7. Impact of lecture-transcript alignment strategies. We also compare our approach with a more popular approach that uses Noise Contrastive Estimation (NCE) loss for aligning video-text pairs [2]. The results are reported in Table 7. Our approach, which uses max-margin ranking loss outperforms the NCE loss perhaps due to the scale of the dataset and the limited number of negative samples in the batch. We were unable to train with larger batch sizes due to GPU memory restrictions.

[2]Multivariate Calculus - segment-1, segment-2, segment-3

Method	NMI ↑	MOF ↑	IOU ↑	F1 ↑	BS@30 ↑
NCE	70.6	71.5	56.3	66.3	43.2
Ours	**79.8**	**80.3**	**69.2**	**76.9**	**58.7**

Table 7: Segmentation performance when lecture-transcript alignment is done using Noise Contrastive Estimation (NCE) loss.

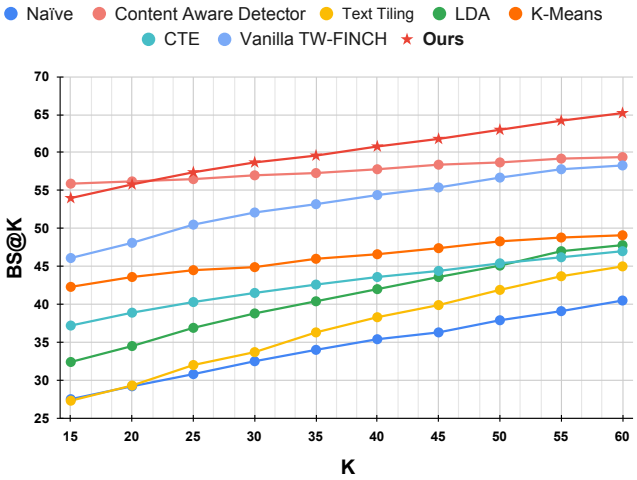

Figure 5: Boundary scores at different values of K.

E. Training details

We train our joint text-video embedding model's parameters with the max-margin ranking loss. We use a mini-batch size of 32. Our model is trained on a 1080ti NVIDIA GPU using Adam optimizer with a learning rate of $1e$-4 and a learning rate decay of 0.9. We use the same hyperparameters for both the pre-training and fine-tuning.

F. Additional Qualitative Results: Retrieval and Segmentation

This section shows additional qualitative results for the text-to-video retrieval and the lecture segmentation task. Fig. 6 shows some of the retrieved clips for different text queries like *graphs coloring*, *operating systems*, etc. We also tested a query *erasing board* to check the model's comprehension of non-conceptual keywords, as shown in the last example of the figure. Although this query is not

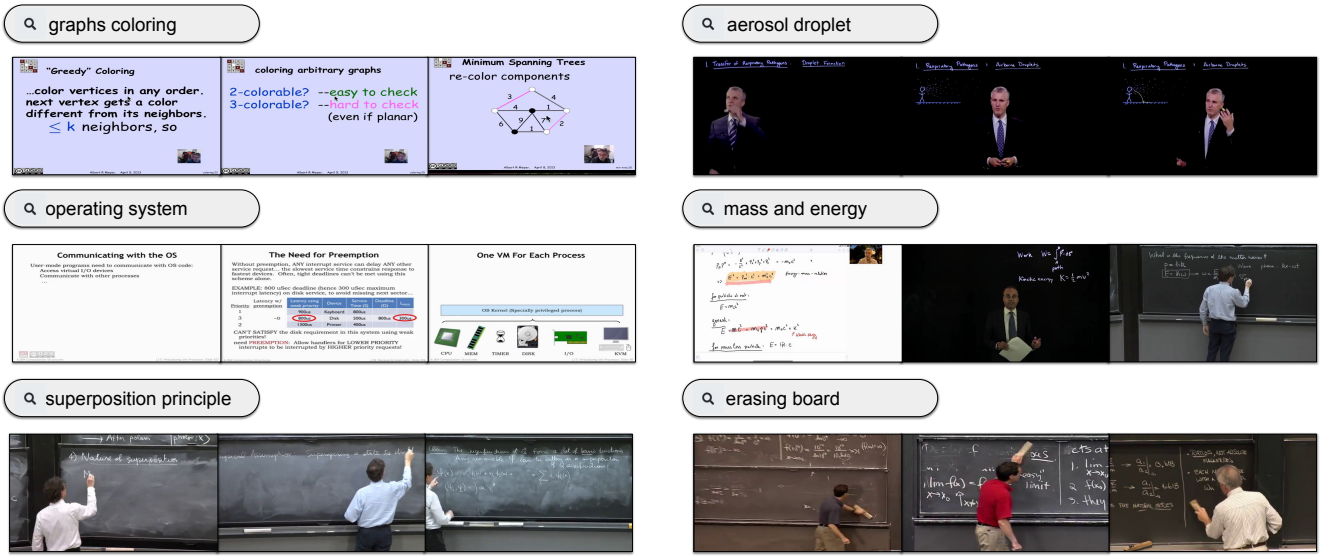

Figure 6: Text-to-video retrieval results on six queries. The figure shows the thumbnails of the top 3 retrieved lecture clips from our model. Our model is able to retrieve relevant lecture clips according to the query.

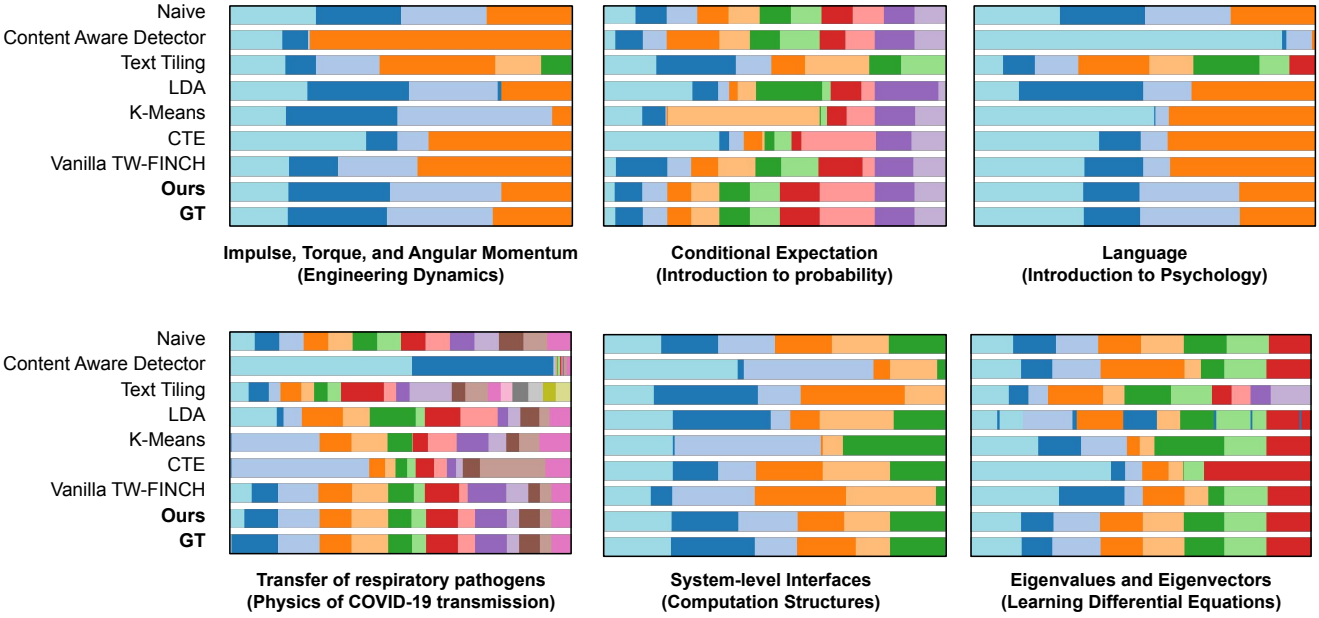

Figure 7: Segmentation examples for six lectures from different courses with varying a number of segments.

present in the transcript, it still correctly retrieves the clips in which the professor erases the blackboard. This demonstrates the importance of pre-training on the CwoS dataset.

Fig. 7 shows more qualitative results from the lecture segmentation task for lectures from different courses. Regardless of the number of segments, our method yields better segmentation length and boundaries when compared with the other baselines.

G. Course-wise segmentation results

We report the top 5 segmentation scores for each of the courses of the CwS dataset across all of its lectures in Table 8. The mapping between the course ID and the course name is shown in Table 9, along with other metadata like the subject area, number of lectures, the average number of segments, and the presentation mode. As seen in the table, our method outperforms all of the other baselines for the majority of the courses. However, there are a few courses

Course ID	Method	NMI	MOF	IOU	F1	BS@30	Course ID	Method	NMI	MOF	IOU	F1	BS@30
	Naïve	72.1	63.9	49.0	61.6	22.2		Naïve	66.3	78.5	66.5	77.1	39.4
	CAD	63.2	55.9	33.4	42.7	29.0		CAD	51.3	66.7	47.7	58.6	35.3
mit001	LDA	70.0	60.7	43.7	55.2	28.1	mit002	LDA	57.2	73.4	58.0	69.5	37.3
	V-TWF	**76.5**	68.2	**52.5**	**62.2**	**45.5**		V-TWF	67.8	77.4	64.7	73.7	54.1
	Ours	**76.5**	**68.4**	52.3	**62.2**	44.2		Ours	**75.0**	**83.9**	**73.9**	**80.6**	**57.3**
	Naïve	72.0	79.0	67.7	78.0	47.3		Naïve	75.8	83.2	72.1	82.8	29.8
	CAD	**96.1**	**96.0**	**94.9**	**94.8**	**94.8**		CAD	**94.9**	**94.0**	**89.8**	**92.1**	**91.4**
mit032	LDA	68.4	75.9	62.0	72.3	50.7	mit035	LDA	70.2	74.0	60.0	72.0	28.0
	V-TWF	78.7	80.4	71.0	78.2	72.4		V-TWF	71.3	70.5	56.6	67.5	39.2
	Ours	87.8	88.2	81.9	86.3	86.5		Ours	77.3	79.6	69.0	78.2	45.7
	Naïve	73.0	82.1	70.9	81.7	26.3		Naïve	70.1	78.9	67.1	77.4	44.8
	CAD	**98.0**	**97.1**	**95.8**	**96.8**	**96.7**		CAD	76.6	77.5	61.8	66.9	70.7
mit038	LDA	69.7	73.9	59.7	71.0	27.2	mit039	LDA	78.2	82.3	69.7	77.7	62.2
	V-TWF	74.6	77.6	64.2	73.6	42.7		V-TWF	76.8	81.2	69.2	77.5	57.6
	Ours	76.7	78.9	66.9	75.8	45.8		Ours	**83.4**	**86.0**	**77.6**	**83.2**	**75.6**
	Naïve	73.0	72.9	58.7	71.4	26.2		Naïve	74.4	**76.0**	**63.1**	**74.5**	30.0
	CAD	**94.3**	**89.5**	**84.3**	**86.5**	**90.1**		CAD	57.7	56.2	35.7	44.8	26.0
mit049	LDA	78.8	79.7	66.4	76.6	47.4	mit057	LDA	68.6	67.1	51.3	62.8	30.6
	V-TWF	82.2	79.8	66.6	74.3	62.9		V-TWF	71.2	69.3	53.7	65.2	32.8
	Ours	84.4	84.7	73.8	81.4	63.2		Ours	**76.3**	**76.0**	62.8	72.4	**41.2**
	Naïve	**74.5**	**77.2**	**65.0**	**76.2**	34.6		Naïve	73.9	72.4	58.7	**71.4**	24.1
	CAD	57.8	57.1	35.6	45.6	24.2		CAD	68.3	57.5	37.4	47.4	42.1
mit075	LDA	74.0	73.8	59.8	70.2	**40.3**	mit088	LDA	76.9	72.2	58.2	68.6	46.0
	V-TWF	72.2	71.5	56.7	67.6	35.4		V-TWF	79.2	71.7	57.2	66.1	54.2
	Ours	73.4	74.8	60.6	71.3	35.2		Ours	**80.3**	**74.8**	**61.8**	71.0	**56.0**
	Naïve	65.7	81.6	70.2	80.4	43.8		Naïve	67.0	66.6	51.1	63.5	21.7
	CAD	63.0	75.3	61.2	68.9	58.7		CAD	52.0	57.9	33.6	42.1	27.5
mit097	LDA	65.8	79.6	66.2	74.8	56.5	mit126	LDA	67.7	68.0	52.5	63.8	24.4
	V-TWF	72.3	81.9	71.5	79.7	67.3		V-TWF	69.0	69.1	51.5	62.1	39.4
	Ours	**79.2**	**86.1**	**77.5**	**83.9**	**72.4**		Ours	**72.4**	**71.1**	**56.4**	**66.4**	**41.5**
	Naïve	76.1	65.2	47.8	60.3	23.3		Naïve	77.1	68.6	53.5	65.5	25.8
	CAD	76.7	66.9	59.0	60.3	63.6		CAD	86.5	76.7	65.1	71.2	70.1
mit151	LDA	77.0	66.9	50.4	61.0	27.4	mit153	LDA	77.7	68.6	50.7	61.1	39.8
	V-TWF	85.2	79.4	64.0	73.2	50.4		V-TWF	85.3	77.5	64.2	72.0	65.4
	Ours	**95.1**	**93.6**	**84.7**	**88.0**	**84.6**		Ours	**90.8**	**84.7**	**74.6**	**79.9**	**78.8**
	Naïve	81.2	85.6	76.4	85.5	31.6		Naïve	71.8	75.5	62.7	74.0	32.5
	CAD	95.8	91.7	88.7	91.0	92.5	Average	CAD	72.9	73.3	59.4	65.9	57.0
mit159	LDA	78.6	76.8	65.9	75.6	31.3	(across all	LDA	70.0	72.4	57.6	68.2	38.8
	V-TWF	81.8	80.9	69.7	77.6	61.2	the 350 lectures)	V-TWF	74.9	75.1	61.7	70.9	52.1
	Ours	**98.4**	**99.4**	**98.8**	**99.4**	**97.2**		**Ours**	**79.8**	**80.3**	**69.2**	**76.9**	**58.7**

Table 8: Course-wise segmentation scores. Here, CAD stands for Content Aware Detector, V-TWF : Vanilla TW-FINCH applied on the concatenation of visual and textual features. The last panel shows the average scores across all the 350 lectures of the CwS dataset.

(mit032, mit035, mit038, and mit049) for which the Content Aware Detector baseline has scores better than the other methods. These are the courses where we combine the individual shorter video segments to form the complete lecture. Since each of these shorter video segments was filmed independently, the lighting/camera angle may have been slightly

Course ID	Course Name	Subject area	# Lectures	Avg. # segments	Mode
mit001	Single Variable Calculus	Mathematics	35	7.9	Blackboard
mit002	Multivariable Calculus	Mathematics	35	3.2	Blackboard
mit032	Classical Mechanics	Physics	38	4.3	Digital Board
mit035	Quantum Physics I	Physics	24	4.8	Blackboard
mit038	Quantum Physics III	Physics	24	4.2	Blackboard
mit039	Introduction to Special Relativity	Physics	12	4.2	Digital Board
mit049	Introduction to Nuclear and Particle Physics	Physics	11	6.1	Digital Board
mit057	Introduction to Psychology	BCS	24	5.5	Blackboard
mit075	Principles of Microeconomics	Economics	26	5.1	Blackboard
mit088	Computation Structures	EECS	21	6.6	Slides
mit097	Mathematics for Computer Science	EECS	35	3.2	Slides
mit126	Engineering Dynamics	ME	27	5.3	Blackboard
mit151	Physics of COVID-19 Transmission	Biology	4	9.4	Digital Board
mit153	Introduction to Probability	EECS	26	9.3	Slides
mit159	Learn Differential Equations	Mathematics	8	6.9	Blackboard

Table 9: Mapping between course IDs and course names along with additional metadata. Here, BCS stands for Brain and Cognitive Sciences, EECS - Electrical Engineering and Computer Science, and ME - Mechanical Engineering.

different for each of these segments. This makes it easier for the Content Aware Detector to predict accurate boundaries. For the other courses, the Content Aware Detector scores are considerably lower than most of the other baselines and our model. All in all, our model outperforms all of the other baselines on an average across all the lectures of the CwS dataset easily, as shown in the last panel of Table 8.

References

[1] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[4] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.