Class Prototypes based Contrastive Learning for Classifying Multi-Label and Fine-Grained Educational Videos (Supplementary Material)

Overview

This supplementary material is organized into four sections. In Section A we provide additional details about our APPROVE dataset. Section B presents detailed classwise result for our models, demonstrating the impact of multi-modal learning. In Section C we analyze the feature space of our trained model to demonstrate that it picks up semantic similarities between the classes. Finally, we provide implementation details in Section D to assist reproducibility. Note that this document contains 12 pages and some are partially blank to clearly separate different parts of the material.

A. APPROVE Dataset

The key property of APPROVE that sets it apart from prior datasets is its fine-grained and multi-label nature. We provide some visualizations here to build on the main paper and illustrate these properties.

A.1. Fine-Grained

Additional samples from the Literacy (in Figure 1) and Math (in Figure 2) splits of APPROVE are visualized here. These examples are randomly picked, and they highlight the fine-grained nature of the dataset. Many pairs or groups of classes in APPROVE have very high visual similarity, e.g. `Shape ID` and `Building Drawing Shapes`; `Sounds in Words` and `Rhyming`, etc. Also note that background, math and literacy are distinct and do not share overlapping labels, which illustrates the heirarchical struture of APPROVE.

A.2. Multi-Label

APPROVE is a densely multi-labelled dataset. The multi-label co-occurence matrix for APPROVE is visualized in Figure 3. Each cell of the matrix, $L[i,j]$, equals the fraction of videos with class i which also contain class j. Note that the matrix is not symmetric as two classes might have a one-sided relationship. e.g. presence of `written numerals` suggests `comparing groups` is highly likely to be present, however the inverse does not hold. since many videos teach how to compare groups without using written numberals, e.g. comparing groups of objects by some non-numeric property such as a shape or color.

A.3. Class descriptions

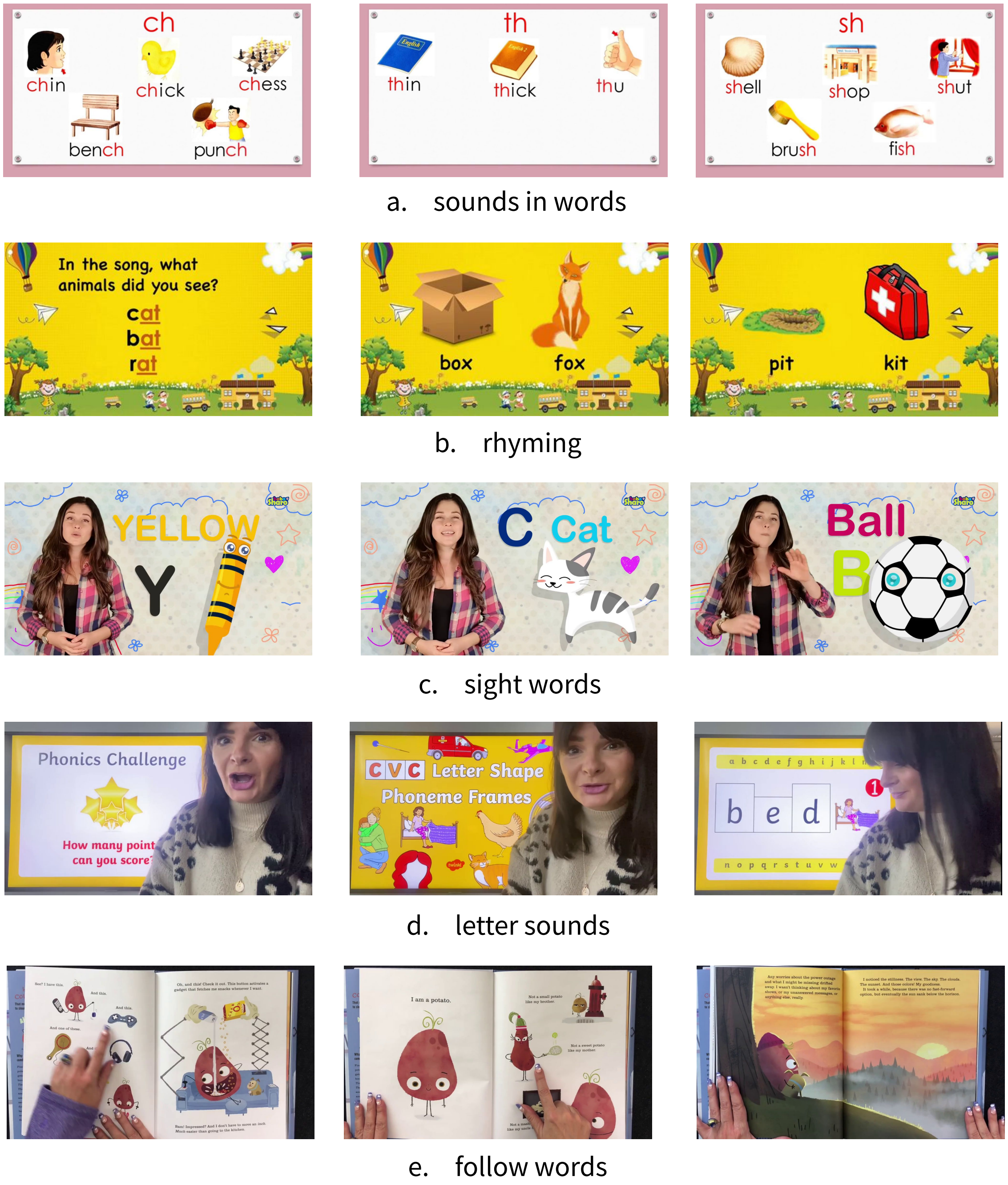
Detailed description of the education codes corresponding to each class in APPROVE and their annotation criteria are presented in Table 1. These fine-grained classes correspond to age-appropriate curriculum topic recommendations prescribed by the common core education standards. These detailed descriptions along with a large batch of examples was provided to all data annotators to ensure high quality labeling.

A.4. Annotation evaluation

To ensure the quality and correctness of the annotations, we consider educational researchers to annotate the videos and follow a standard validation protocol [6]. We consider two expert annotators and a few education researchers for annotating the videos. Experts train annotators to identify the indicators of educational content in videos. After training, education researchers are evaluated on a validation set of 50 videos that are already annotated by two experts. Annotators are allowed to start the final annotation process once they achieve more than 90% agreements with the expert annotations. We observe a 95% agreement is reached after four weeks of training. We also consider inter-annotator consistency for the final annotations. Videos that are not consistently labeled by all the annotators are ignored.

A.5. Education codes

APPROVE consists of 193 hours of videos with 19 classes including 7 literacy codes, 11 math, and background. We follow the Common Core Standards [2, 5] to select education content suitable for the kids at kindergarten level. Descriptions of these codes are provided in Tab. 1. Some sample frames for these codes are presented in Fig. 1 and 2.



a. sounds in words

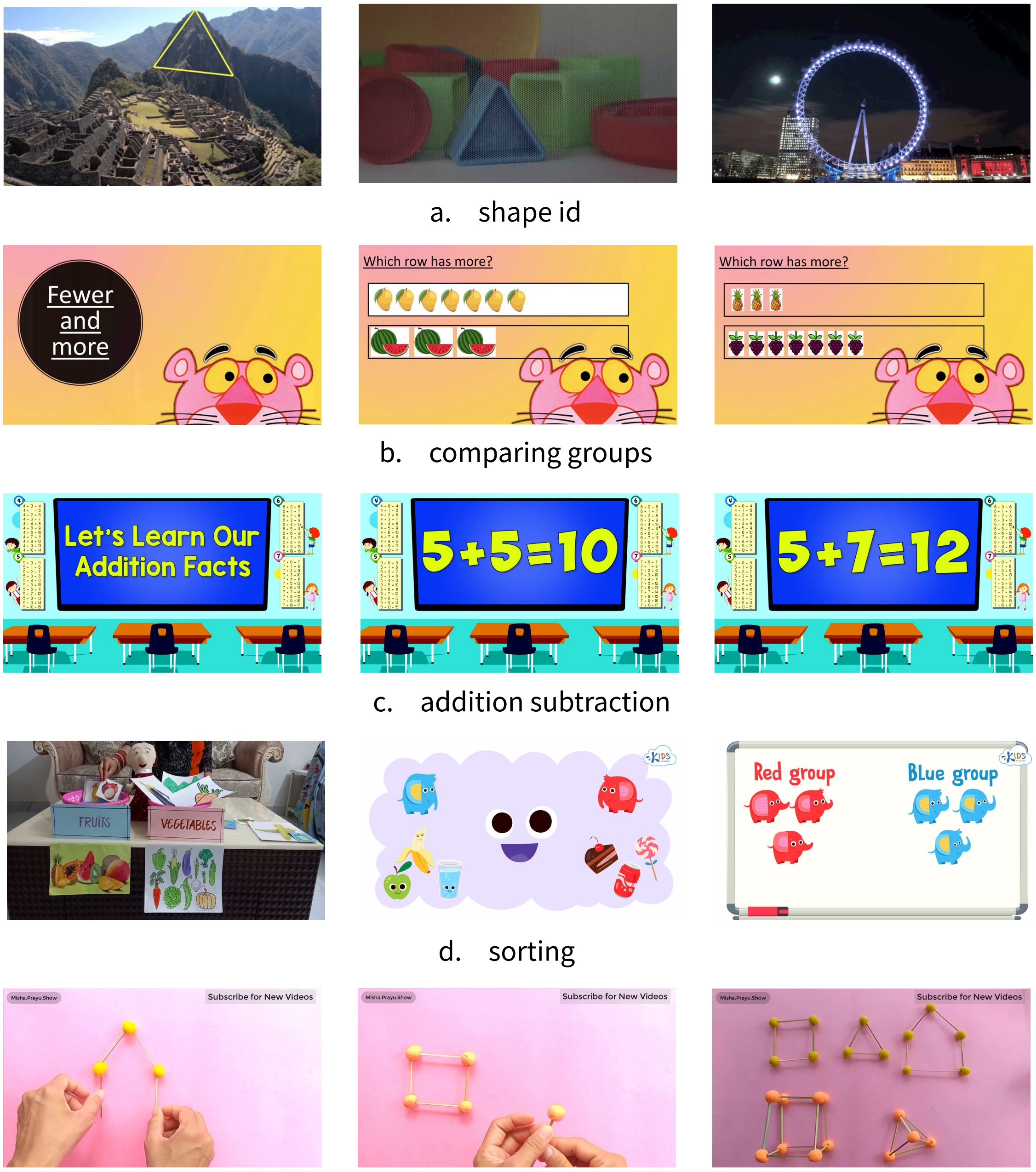
b. rhyming

c. sight words

d. letter sounds

e. follow words

Figure 1. Sample frames from five Literacy classes in APPROVE. The classes share visual similarity, which makes classification a challenging fine-grained learning task.



a. shape id

b. comparing groups

c. addition subtraction

d. sorting

e. building drawing shapes

Figure 2. Sample Frames from five Math classes in APPROVE.

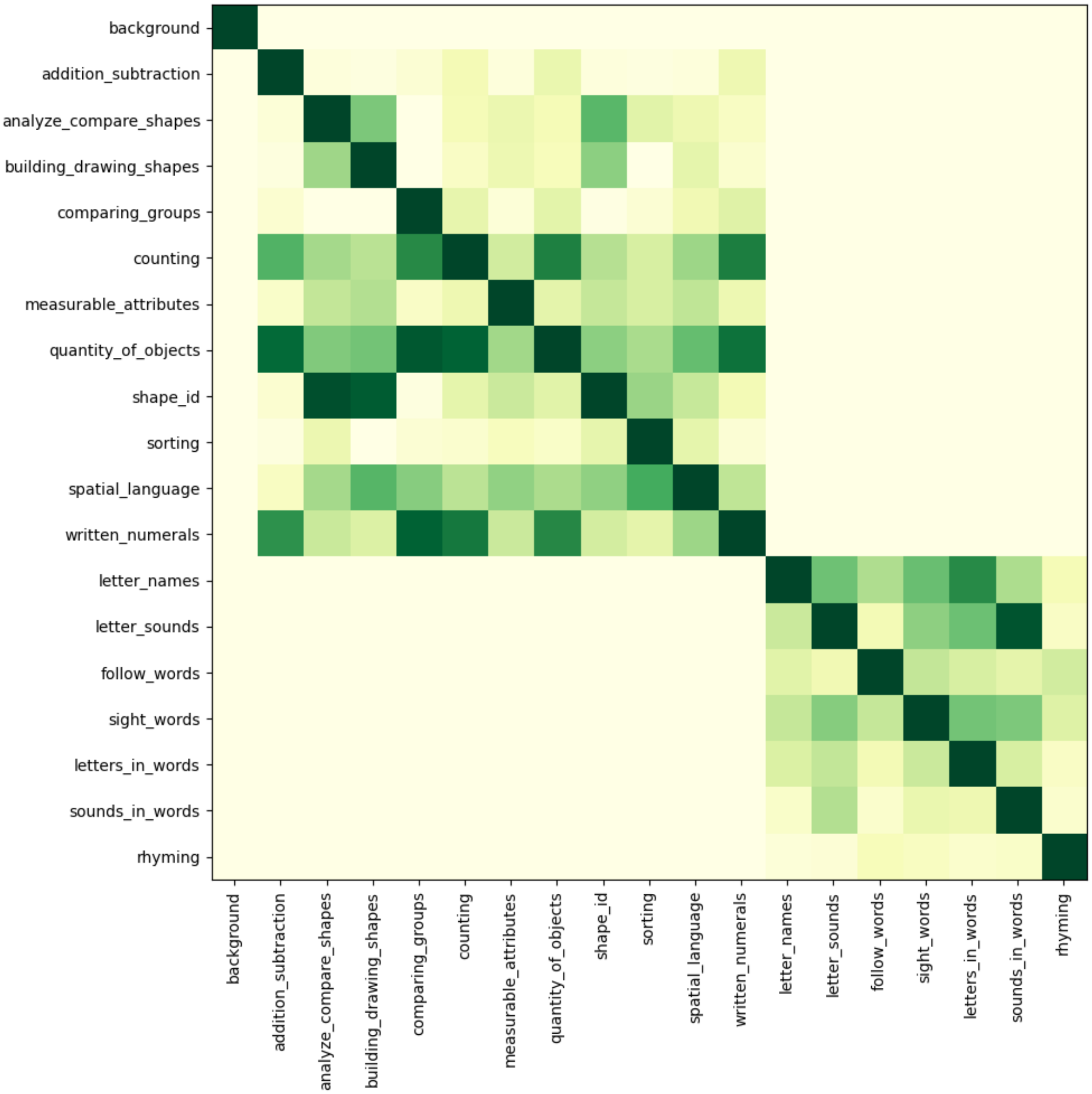


Figure 3. Ground=Truth Multi-Label co-ccurence matrix for APPROVE videos. The three high level groups of categories: Background, Math and Literacy can be seen, highlighting the hierarchical structure of the dataset.

Table 1. Description of the education codes used in APPROVE. These codes correspond to age-appropriate curriculumn recommendations prescribed by the common core education standards.

Code names	High-level class	Description of the code
Counting	Math	• More than one number in the standard sequence. Starting point can be any whole number. • This includes counting parts of a shape, such as counting sides or vertices.
Written numerals	Math	• A written numeral, either on its own or as part of a count sequence, with corresponding visual or audio.
Quantity of objects	Math	• Emphasizing that the last number in a count sequence represents the total number of objects.
Addition or subtraction	Math	• At least two numbers and the number that results when they are added or subtracted. • Includes adding by counting on. For example, "We have three, let's add two. Three, four, five. Three and two make five." • Includes decomposing sets of objects into two or more sets.
Measurable attributes	Math	• Describing one object or comparing multiple objects based on at least one measurable attribute, such as length or volume.
Comparing groups	Math	• Comparing two or more groups of objects.
Sorting	Math	• Objects being sorted into categories, such as but not limited to color, shape, object type, purpose, pattern, or species.
Spatial language	Math	• Words and visuals to describe position or movement.
Shape identification	Math	• Naming and displaying a shape.
Building or drawing shapes	Math	• Showing how a geometric shape is drawn or built.
Analyzing or comparing shapes	Math	• Describing one or more shapes in terms of their attributes.
Letter names	Literacy	• Any spoken or sung letter name. • Does not need to say whether the letter is upper- or lowercase.
Letter sounds	Literacy	• Any spoken or sung letter sound. Must be distinct from a spoken word. • Can include letter names if they are also letter sounds (e.g., long forms of vowels). • Does not need to say whether the letter is upper- or lowercase.
Letters in words	Literacy	• A letter within a word is visually highlighted. • If the video names multiple letters within a word, to meet this criterion it must highlight each letter individually as it is named. • If the video separately names the letter and then displays it in a word, the letter must be visually highlighted within the word.
Sight words	Literacy	• Only words on the sight words list count for this code. As long as a video includes at least one word on the sight words list, this indicator is present.

Continued on next page

Table 1 – continued from previous page

Code names	High-level class	Description of the code
Sounds in words	Literacy	• The sound may occur anywhere within a word, including the beginning sound. • The sound must not be the full word. • Choose response option, sound of individual letter if the video includes audio of the sound a single letter makes within a word. • Choose response option, sound of multiple letters together if the video includes audio of the sound of two or more letters together within a word, excluding the full word. For example, “at” in “rat.” • Can include separately making the sound of a letter on its own and displaying it in a word, so long as both occur within about 2 seconds. • Still counts even if other words are used between the full word and the sound. For example, “The words cat and rat both have the ‘t’ sound at the end.”
Follow words	Literacy	• Must show a passage containing multiple words. Only one word on screen at a time would not count. • Words must be highlighted left to right, top to bottom, and/or page to page. Highlighting can include one word in a passage appearing at a time. • It’s okay if words aren’t highlighted exactly as they are spoken (e.g., highlighting an entire line of text in a paragraph at a time, highlighting words at a constant pace that doesn’t totally line up with audio), so long as the highlighting generally moves left to right or top to bottom as the words are spoken. • Includes sing-along style videos that highlight words as they are sung.
Rhyming	Literacy	• Within 60 seconds of the word ”rhyme” or ”rhyming,” audio of at least 2 rhyming words. • “Rhyme” may occur before or after the rhyming words. • Rhyming words do not need to be spoken one after the other (e.g., “cheese, please”); they could have words between them, such as a poem or song (e.g., the cat jumped over the hat).

End of Table 1

B. Classwise Results

We demonstrated strong overall results in the main paper. In particular we found that using multi-modal input data resulted in strong results. Here, we provide class-wise recall and F1 scores in Figure 4 and Figure 5 respectively. These show that our improvements occur across a wide variety of video classes. In order to compute Recall and F1 score, we pick the classification threshold to achieve 80% overall precision to satisfy the requirements of the sensitive education application scenarios (as discussed in the main paper). The threshold found are 0.91 for Video only model, 0.56 for the Text only model and 0.51 for the Video+Text model. As can be noticed in Figure 4 Text only model generally outperforms the Video only model, but the Text+Video model outperforms the Text only model for most classes. Classes which focus on skills requiring connecting language to vision such as `Sight Words`, `Written Numerals` and `Sorting` benefit the most from the use of multi-modal data for classification.

In Figure 6 we provide a scatter plot of classwise recall for the text only model recall vs the video only model. The recall is weakly correlated across the two modalities ($R^2 = 0.122$), which explains the significant gains due to combining the two modalities.

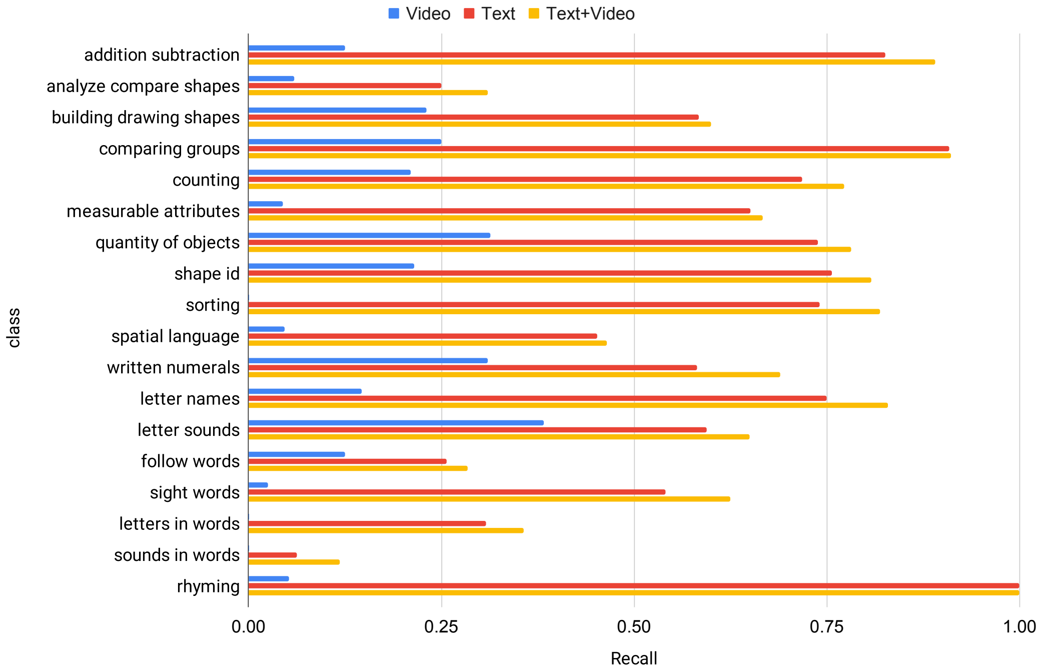


Figure 4. Classwise Recall at 80% overall Precision. Most classes benefit from access to multi-modal input data.

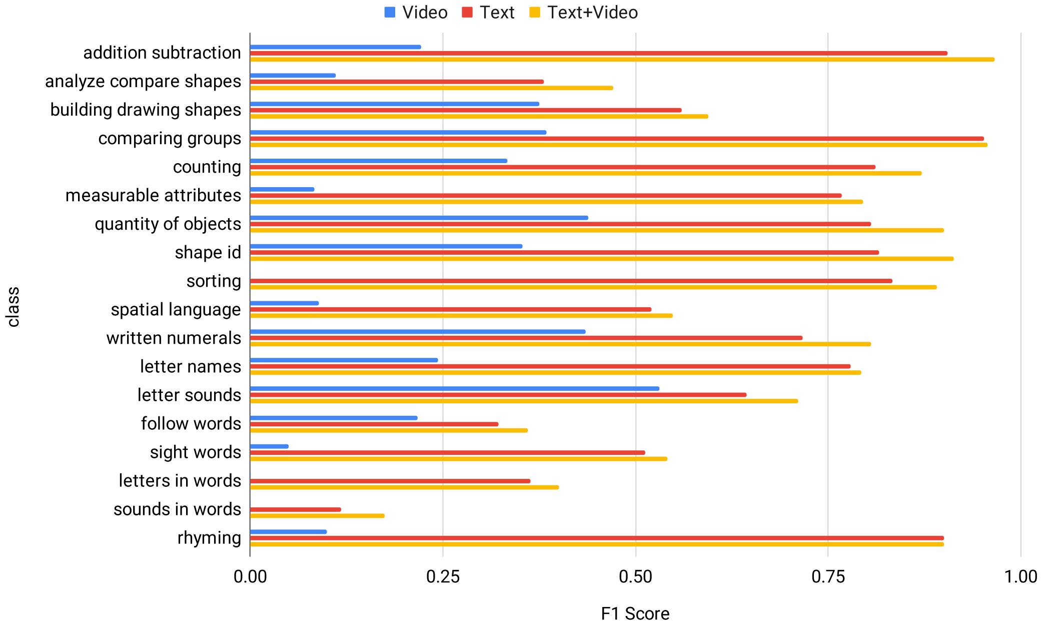


Figure 5. Classwise F1 Scores at 80% overall Precision.

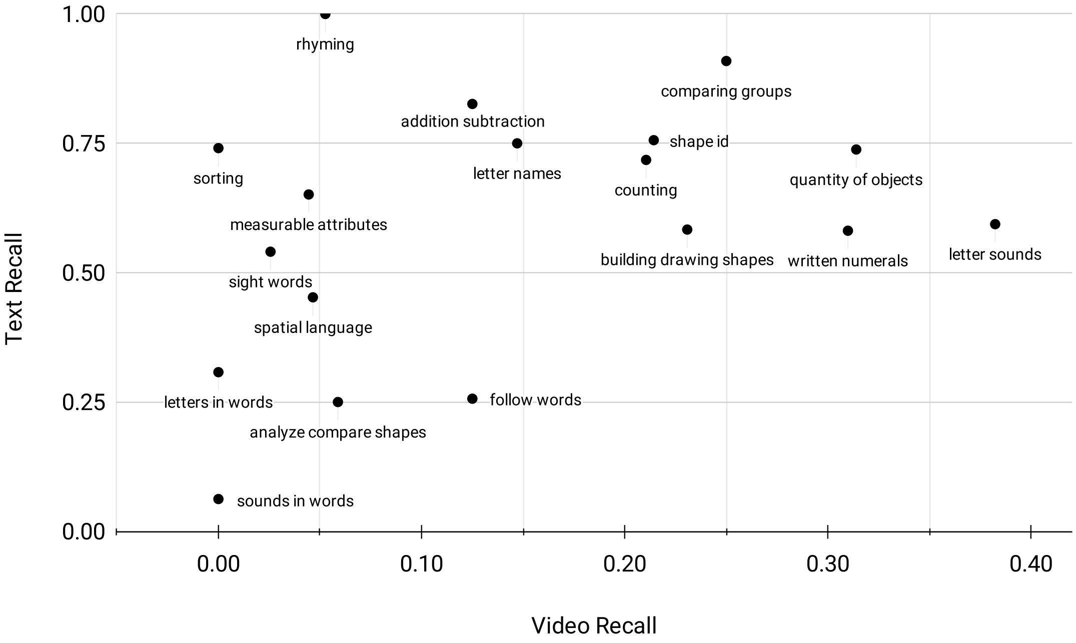


Figure 6. Comparing classwise recall between video and text models.

C. Learned Feature Representation

We visualize learned features from our model using t-SNE in Figure 7. At the top level literacy and math videos are cleanly separated. We also plot one-vs-all plots for each class as APPROVE is a multi-label dataset. It can be observed that even fine-grained classes are well clustered, especially for math topics.

D. Implementation Details

D.1. Models

Image Backbone: We use ImageNet pretrained ResNet50 [4] and Instagram user generated tag weakly supervised (SWAG) + ImageNet finetuned ViT-B/16 (384×384 resolution) [3] from the TorchVision model zoo. We also use the ImageNet-21K pre-trained and ImageNet finetuned ViT-B/32 (224×224 resolution) from HuggingFace.

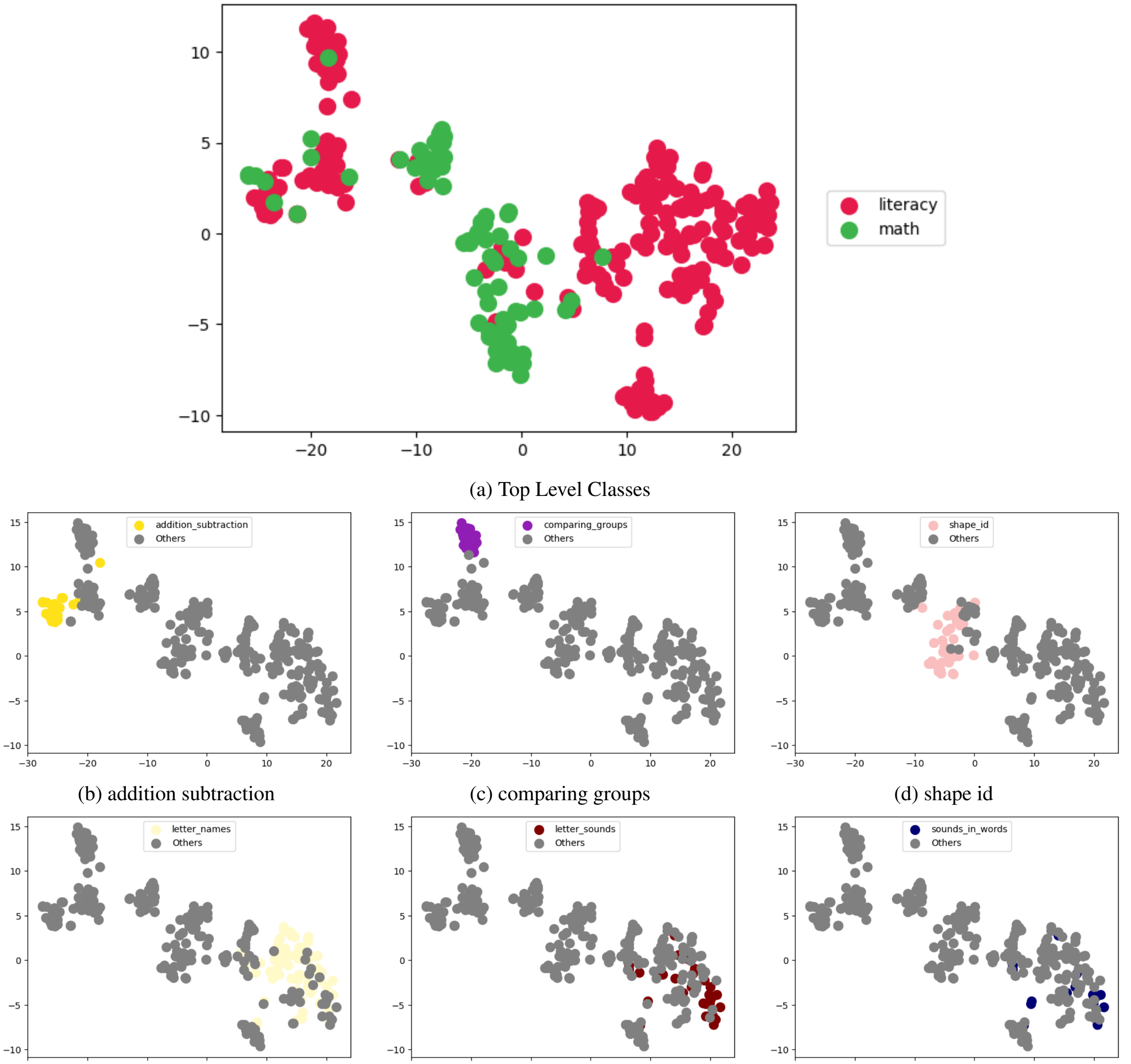
Text Backbone: For the language encoder, we use DistilBERT-Base-uncased [9], and t5-small [8] from the huggingface `transformers` library.

D.2. ASR Generation:

We generate ASR text from the videos using OpenAI-whisper [7], which is an open source ASR model. We used the `medium.en` version of the model and turned off the `condition_on_previous_text` option as we observed that ASR generation would collapse for videos with poor audio quality with that option turned on by default.

D.3. Other Datasets

Some video classification datasets have been proposed with class labels based on *topics* being discussed or illustrated. One such dataset is COIN [10], which consists of instructional videos from 180 diverse coarse-grained tasks covering a wide range from `changing-car-tire` to `making-pizza`. While temporal sub-task segmentation labels are also available, in this paper we restrict ourselves to fine grained video classification task. YouTube-8M(YT-8M) [1] dataset is a large sample of YouTube data labeled with many coarse-grained visual entities, however, because of its large size, its distributed in the form of extracted visual and audio features. In order to fully test the potential of our method on this dataset, we create a subset using 1% of YT-8M data called YT-46K, which consists of 46,000 videos (note that despite its name the full YT-8M only contains 5.6 Million videos, since we scraped a 1% shard, we attempted scraping about 56,000 videos, of which about 46,000 were still available) with full video, audio and text metadata scraped from YouTube. Since it is a long tailed multi-label dataset, the frequency of labels follows a power-law distribution. We restrict the number of classes to those with at least 100 instances, which results in 166 usable labels.



(a) Top Level Classes

(b) addition subtraction

(c) comparing groups

(d) shape id

(e) letter names

(f) letter sounds

(g) sounds in words

Figure 7. Visualizing features for APPROVE test set samples using tsne. **(a)** Top level classes, math and literacy, are disjoint in feature space. **(b-g)** Since APPROVE is a multi-label dataset, we show one-vs-all tsne plots.

D.4. Data Augmentations

We use `RandAugment` and `RandomResizedCrop` for augmenting video frames. RandAugment magnitude is ramped up from 1 to 10 over first 20 epochs.

```python

import torchvision.transforms as t

t.Compose([t.RandAugment(magnitude=magnitude),
           t.RandomResizedCrop(224, scale=(0.4, 1.0), \
           interpolation=t.InterpolationMode.BILINEAR),
           t.ConvertImageDtype(torch.float32),
           t.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)),
           ])

```

For augmenting text datat we use synonym replacement from paraphrase dataset, random span cropping and random word swapping. As Back Translation is computationally expensive, we compute 4 additional back translated versions of each text before training.

```python
import nlpaug.augmenter.word as naw
import nlpaug.flow as naf

# m represents the magnitude of augmentation
m = 0.5
t = [naw.SynonymAug(aug_src="ppdb", aug_min=2, aug_max=15, aug_p=m)]
t += [naw.RandomWordAug(action="crop", aug_min=1, aug_max=5,  aug_p=m)]
t += [naw.RandomWordAug(action="swap", aug_min=1, aug_max=5, aug_p=m/2.)]
train_augmenter = naf.Sequential(t)


from nlpaug.augmenter.word \
           .back_translation import BackTranslationAug as BTAug

# actual arguments to BTAug are from_model_name, to_model_name
# abbreviated to fit the command in one line
BTAug(from='facebook/wmt19-en-de', to='facebook/wmt19-de-en')
BTAug(from='Helsinki-NLP/opus-mt-en-de', to='Helsinki-NLP/opus-mt-de-en')
BTAug(from='Helsinki-NLP/opus-mt-en-nl', to='Helsinki-NLP/opus-mt-nl-en')
BTAug(from='Helsinki-NLP/opus-mt-en-fr', to='Helsinki-NLP/opus-mt-fr-en')
```

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. YouTube-8M: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 11

[2] National Governors Association et al. Common core state standards. *Washington, DC*, 2010. 2

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 11

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 11

[5] Andrew Porter, Jennifer McMaken, Jun Hwang, and Rui Yang. Common core standards: The new us intended curriculum. *Educational researcher*, 40(3):103–116, 2011. 2

[6] J. S. Radesky, A. Schaller, S. L. Yeo, , H. M. Weeks, and M. B. Robb. Young kids and youtube: How ads, toys, and games dominate viewing. common sense media. Technical report, Common Sense Media, 2020. 2

[7] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. Technical report, Tech. Rep., Technical report, OpenAI, 2022. 11

[8] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020. 11

[9] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 11

[10] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 11