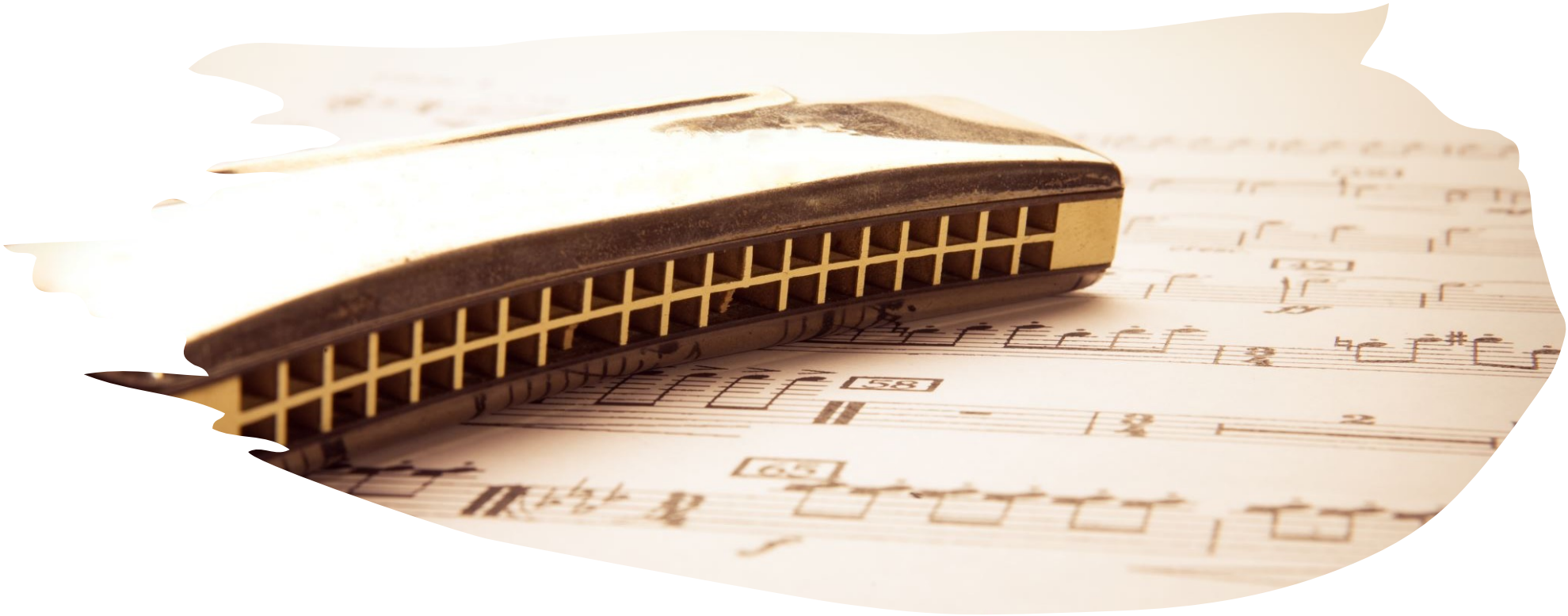


Trimodal Representation of Music

Sagar Joshi

Agenda

- Introduction
- Literature Study
- Dataset Preparation
- Model Architecture
- Pretraining Objectives
- Training & Evaluation
- Concluding Remarks

Agenda

- Introduction
- Literature Study
- Dataset Preparation
- Model Architecture
- Pretraining Objectives
- Training & Evaluation
- Concluding Remarks

About

- Enabling the understanding of music from all the three modalities involved:
  - text
  - audio
  - video
- Developing representations for each of the three modalities in a unified space
- Leveraging pretrained modality-specific architectures for adaptation to the music domain
- Utilizing the signals obtained from large, unlabeled music data sources

Motivations

- Dearth of literature in understanding of music videos through computational methods
- Lack of available work dealing with music from the trimodal perspective
- Few multimodal works in music being bi-modal in nature:
  - audio-text
  - video-text
  - audio-video
- Lack of works exploiting powerful representations from strong pretrained models for music adaptation

Contributions

- A dataset of high-quality YouTube videos spanning 45 different categories in music
- A trimodal architecture for music understanding, that provides representations for text, audio & video in a common space
- Comparison of pretraining strategies for building music representation through their illustration on downstream objectives

Agenda

- Introduction
- Literature Study
- Dataset Preparation
- Model Architecture
- Pretraining Objectives
- Training & Evaluation
- Concluding Remarks

Examples of prior works in music video analysis

- Identification of videos into 23 content categories such as visual abstraction, sex, dance, violence, crime, etc.
- Nature of visual set-up in music videos: mostly identified to be unrealistic and abstract
- Unusual event structures in music videos
- Social implications of high-impact content such as MTV videos on youth and demographic segments
- Evolution of the content in music videos over years: from advent of MTV in 1980’s to proliferation of YouTube content in 2010’s

Examples of prior works in music video analysis

- Music videos shaping youth behavior, being a primary medium in 1980s: introducing youth counterculture
- What do the songs “mean” being a reason for watching music videos among adolescents
- Music videos re-imagined in 2010’s with ”Here It Goes Again” introduced on YouTube
- Viral music videos: novel, succinct, catchy, memorable concepts & visual hooks
- Demographic preferences in music video viewing

Examples of prior works in music video analysis

- Analysis of motivations behind video viewing by different demographic groups
- Storytelling through music videos
- Portrayal of race and sexuality in music videos: analyzing music videos from race or gender-specific distributions
- Characteristics of lead characters in music videos
- Dominance of racial groups across different genres of music videos
- Tobacco and alcohol use behaviors in music videos

Comments on the music video literature study

- All the works involve human analysis on a video set

- Limitations of the experiments & study:
  - Reproducibility
  - Scalability to large number of videos
  - Extensibility to new videos coming with time

- Not all, but most works can greatly benefit from computational models capable of some understanding of the content in the videos

What do we look for?

- Models capable of understanding the music, language and visual parts in music videos
- To start with, we look for models that can represent the three modalities in the same space
- A basic usage of these tasks is multi-modal retrieval: using the input from one modality to retrieve the other
  E.g.
  - Finding relevant music audios or video pieces from textual description
  - Using a music piece (audio) to find out suitable video pieces for that
- More advanced tasks like generation can stem from this development

Agenda

- Introduction
- Literature Study
- Dataset Preparation
- Model Architecture
- Pretraining Objectives
- Training & Evaluation
- Concluding Remarks

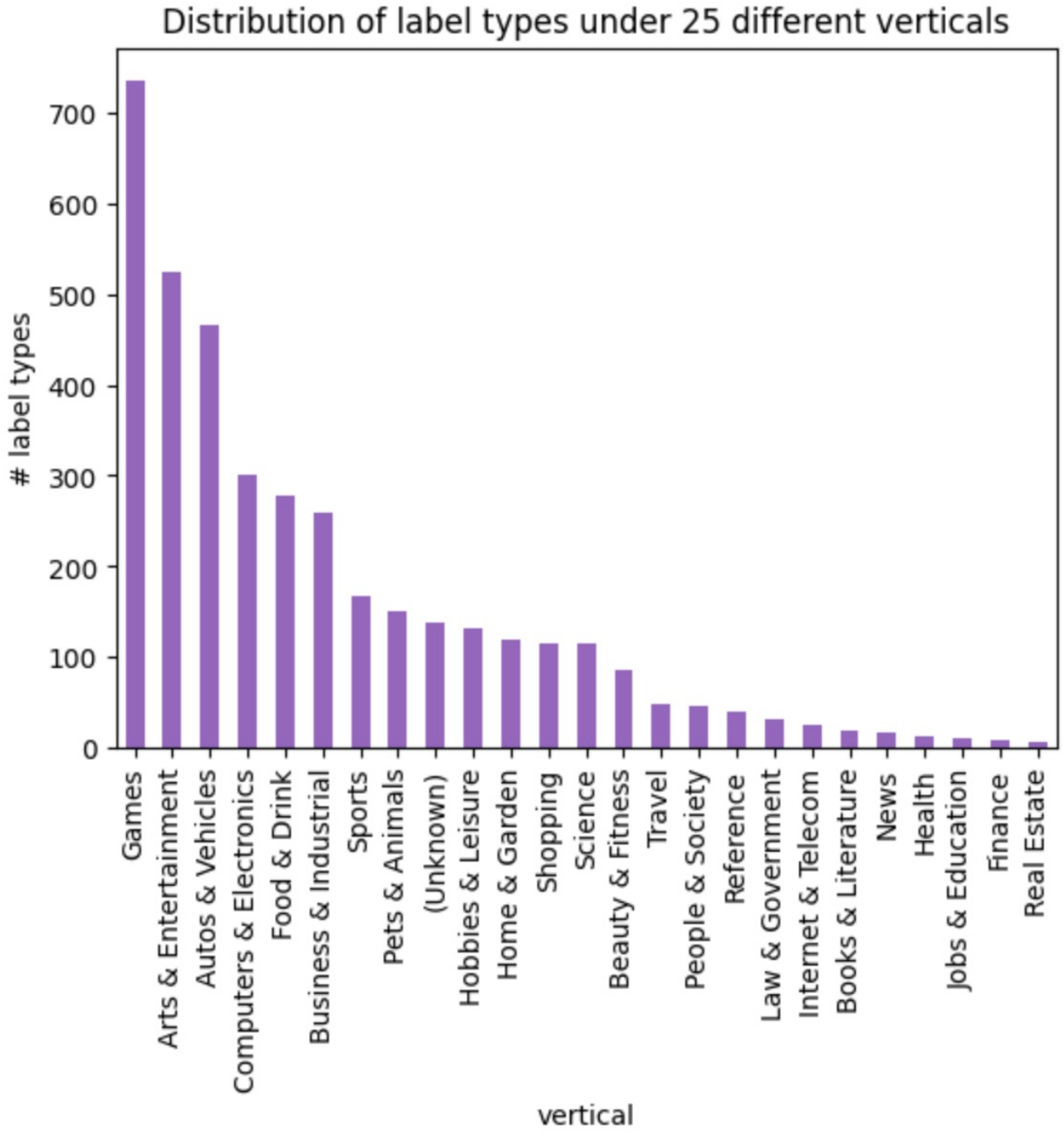
Dataset for Experimentation

- We make use of the YouTube 8M: A large-scale dataset for multi-label classification of YouTube videos
- The dataset consists of approx. 8 million videos (500k hours) from diverse content domains, annotated across 4800 labels, under 25 high-level verticals

Distribution of video labels under 25 verticals

The vertical of interest, “Arts & Entertainment” had

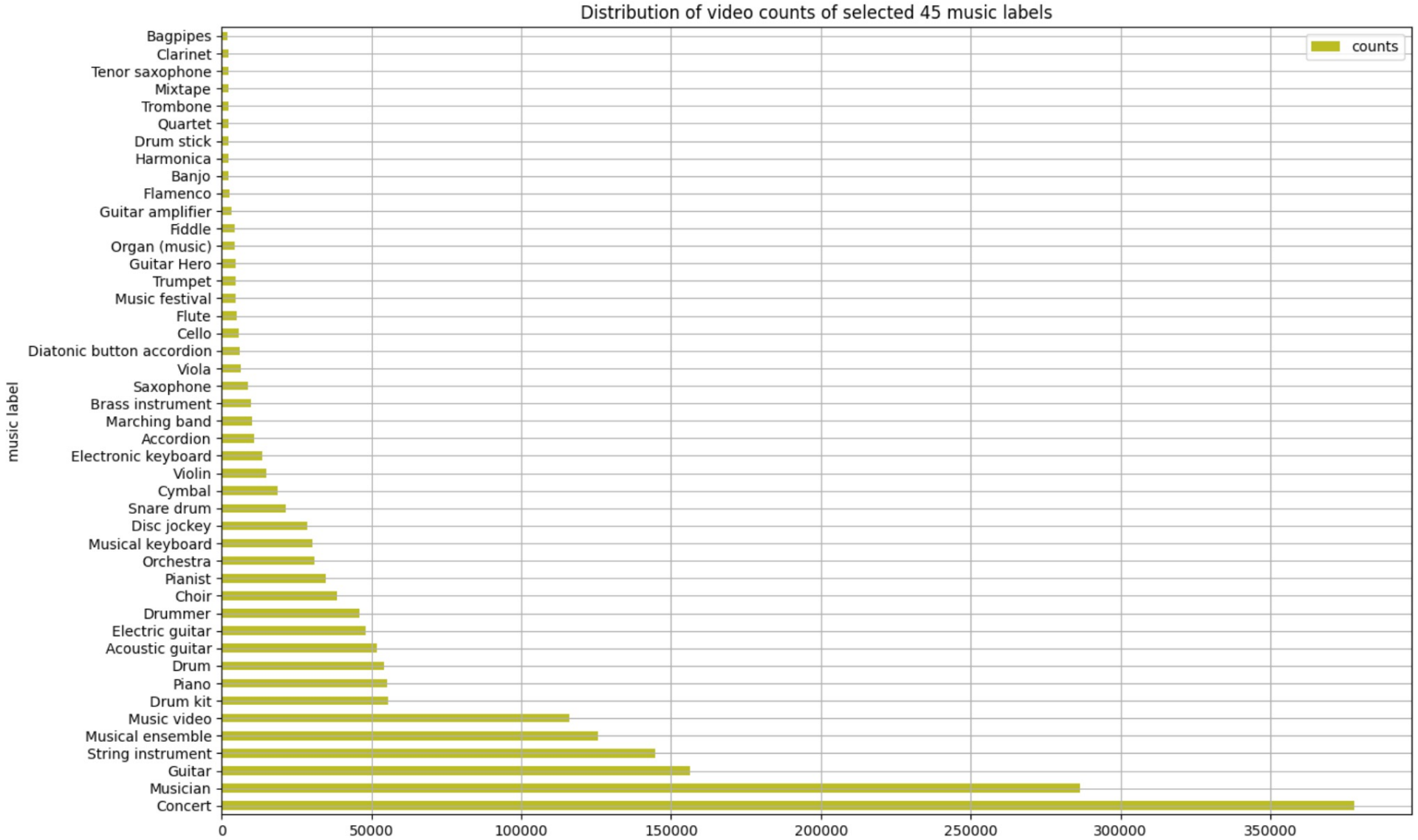
- 525 label types
- > 3.27 million videos



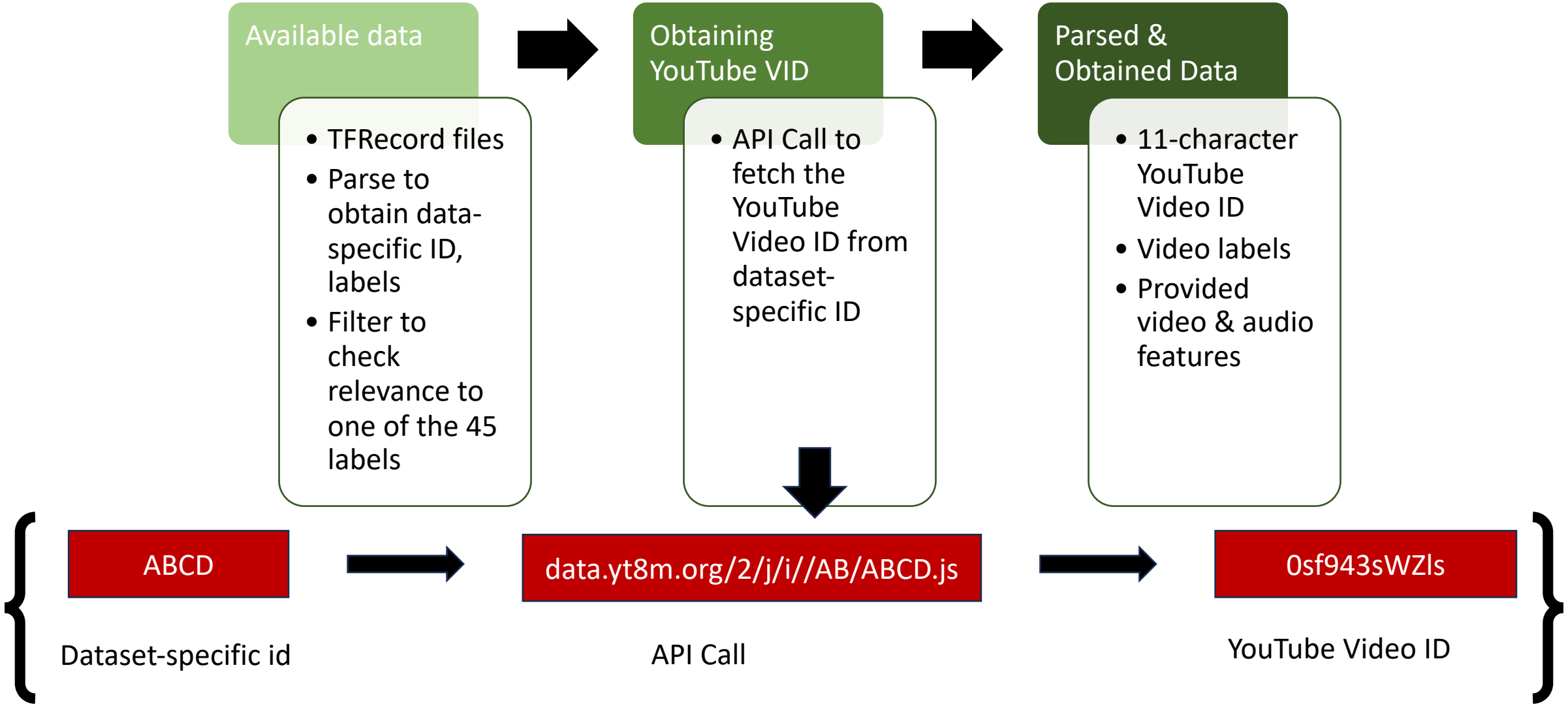
Identification of Music-Related Labels

- Of all the 25 verticals, “Arts & Entertainment” had labels relevant to music
- The labels under this vertical were manually filtered to select top **45 labels** deemed to be relevant to the music domain
- For further steps in dataset preparation, only those videos were considered that belonged to at least one of the 45 categories

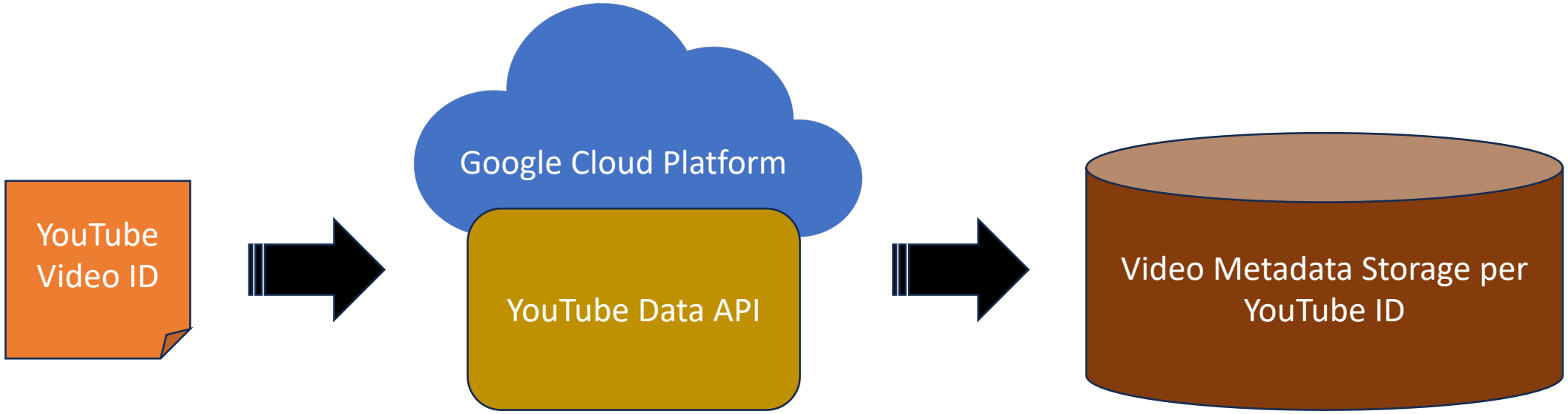
Video Count Distribution for the Selected Labels



Obtaining Labels & YouTube Video Ids from Dataset



Fetching Video Metadata From YouTube Video IDs



- Free tier of the YouTube Data API allows only a maximum of 10,000 req./day
- Hence, data was fetched in batches from the Google Cloud Platform service
- In total, metadata information was fetched for 30,000 video ids in 3 batches
- Dev split of the YouTube 8M dataset was utilized for this

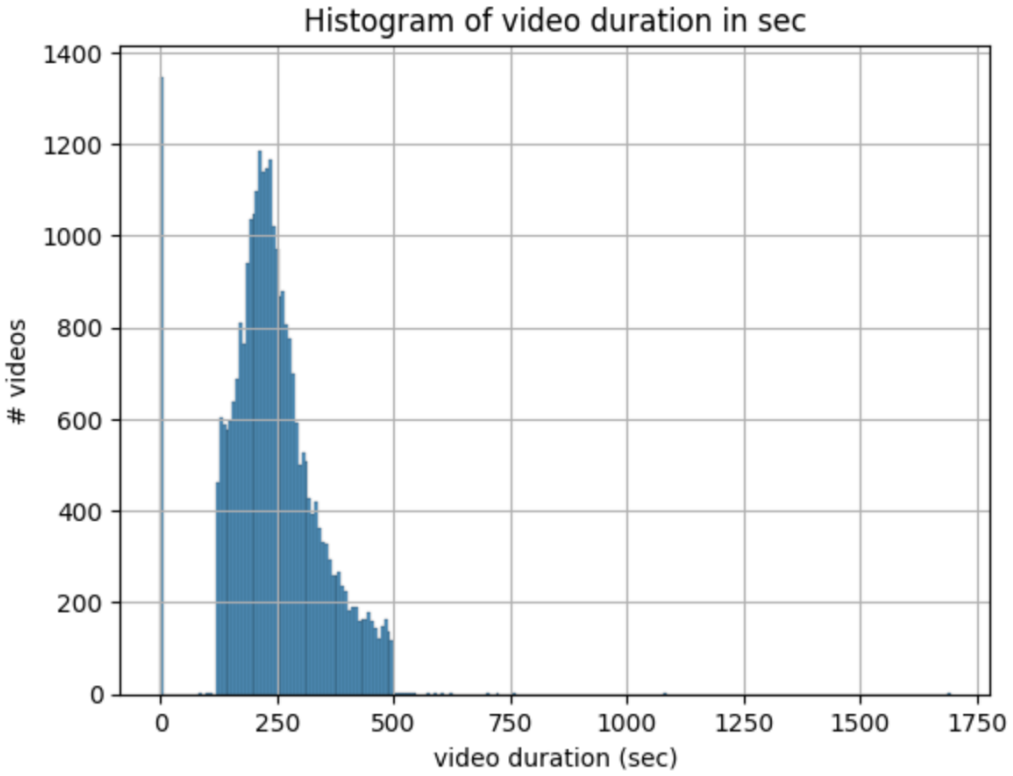
Information Obtained Per Video

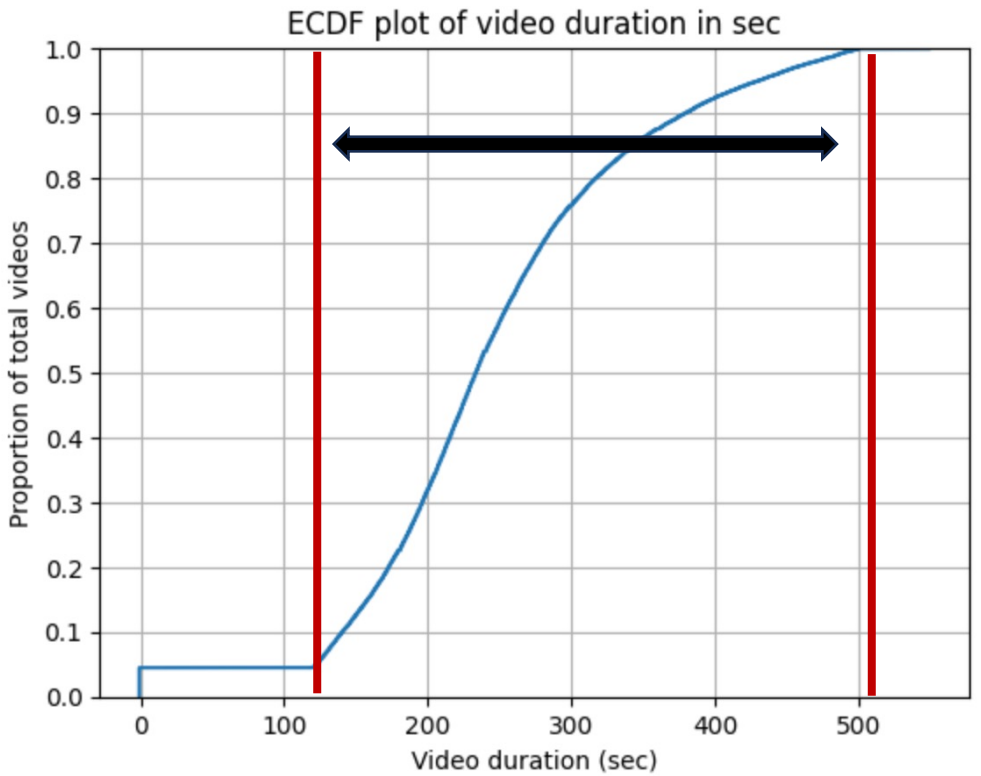
- Title
- Description
- Publish time
- Channel info (channel id, channel title)
- Tags
- Duration
- View count
- Like count
- Comment count
- Quality definition
- Licensing info
- Content rating
- Projection
- Dimension
- Caption

Usage of the Downloaded Metadata

- Studying the nature of the dataset: length, age and quality of the videos (gauged from view/like count)
- Obtaining thresholds to take a subset of the data for training our models on
- Usage of the information such as tags useful to act as textual modality in our trimodal problem

Criteria 1: Video Duration

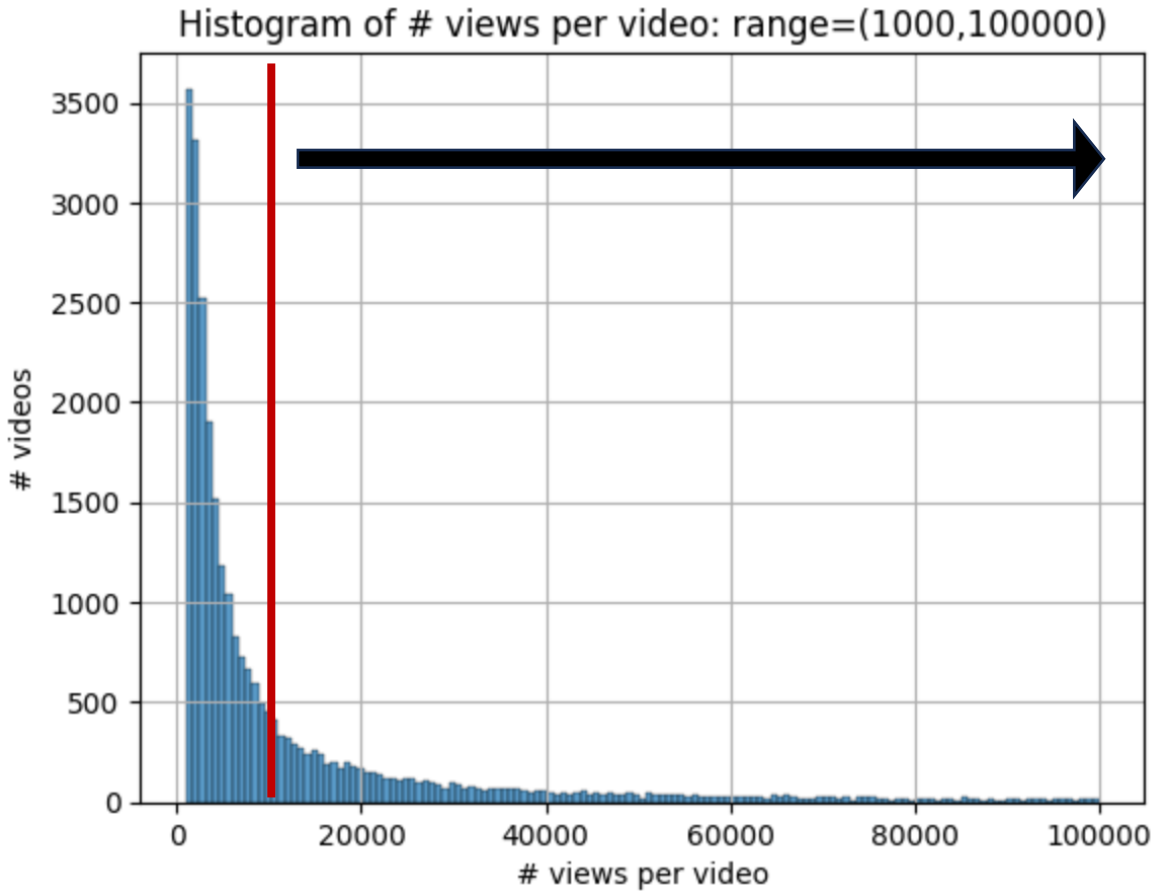




- Minimum duration: 100 sec
- Maximum duration: 512 sec

Duration Stats
Mean: 254 sec
Median: 238 sec

Criteria 2: View Count

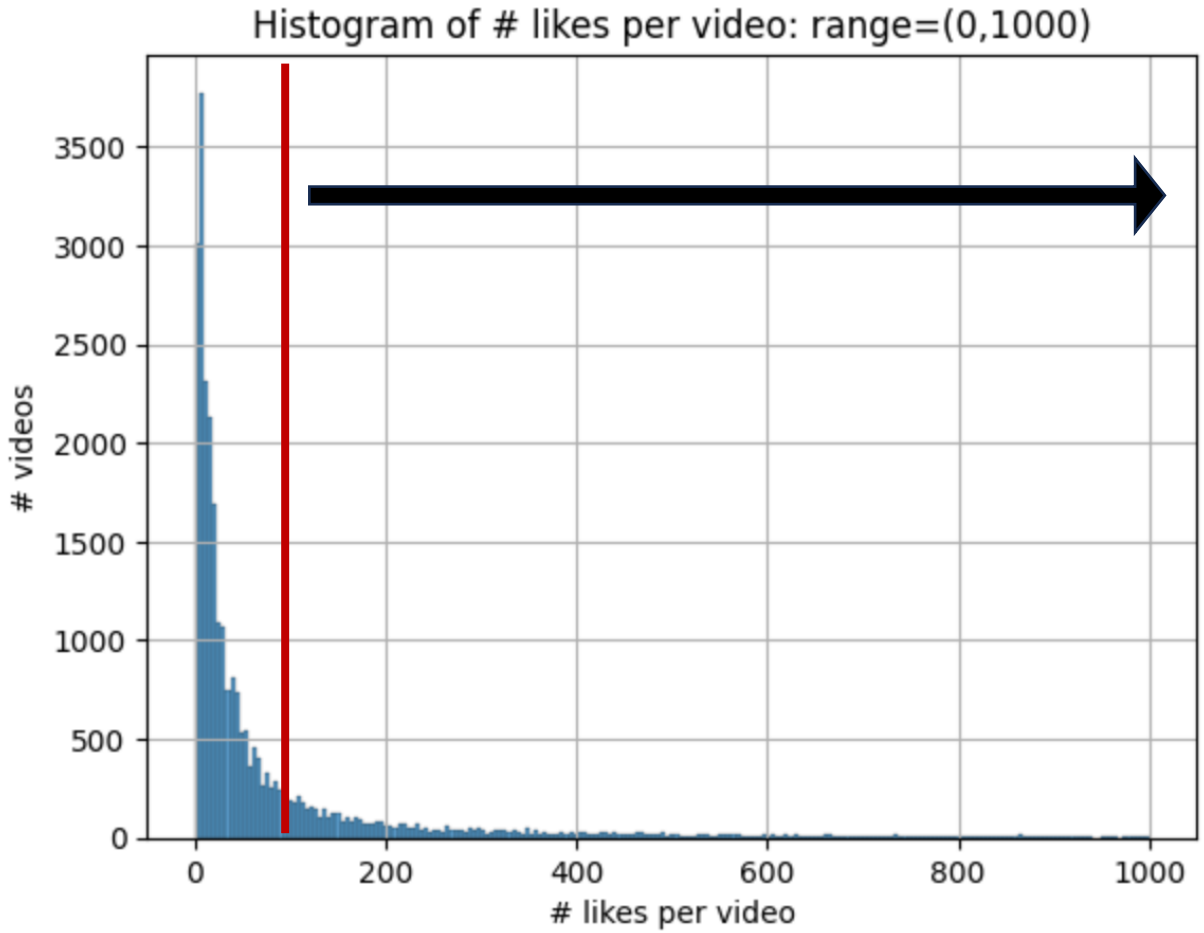


Minimum # views: 10,000

View Count Stats

Mean: 446,728
Median: 30,802

Criteria 3: Like Count

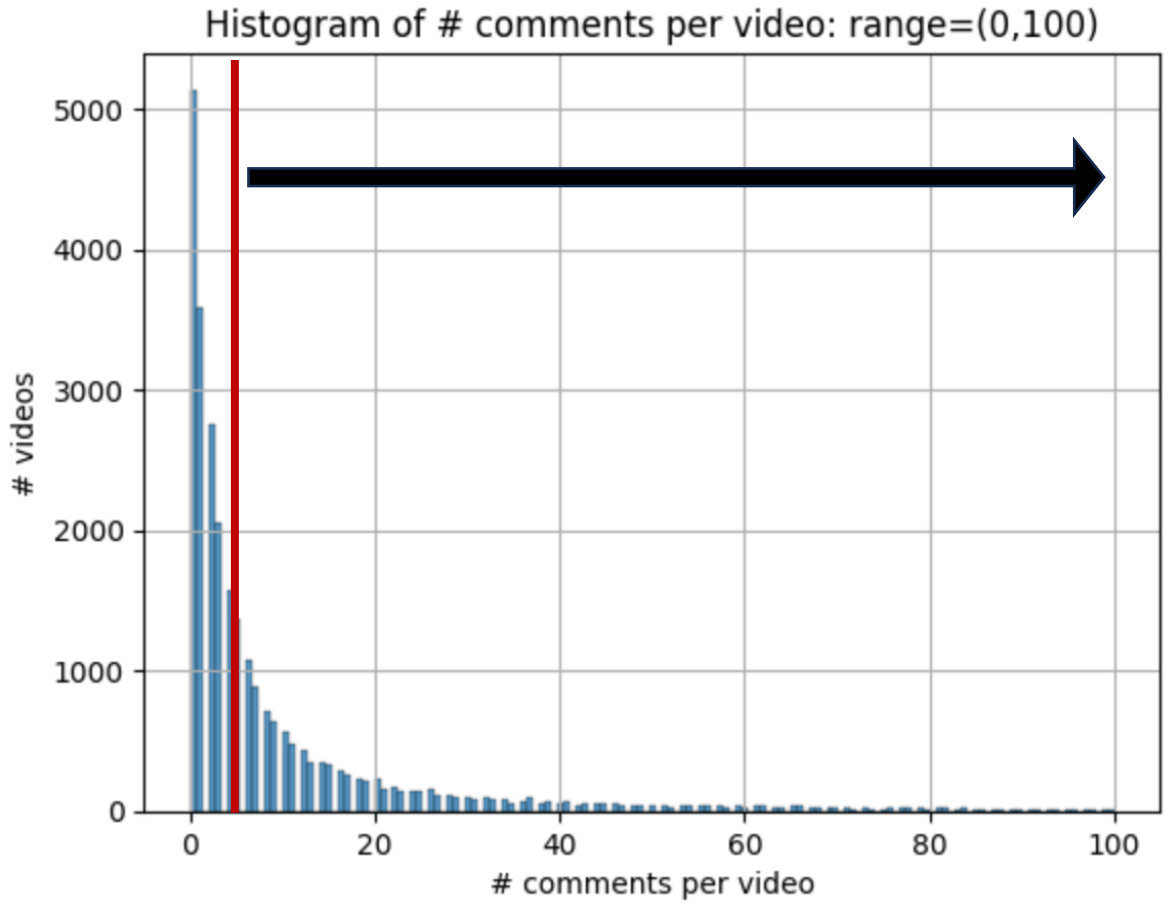


Minimum # likes: 100

Like Count Stats

Mean: 3,858
Median: 342

Criteria 4: Comment Count

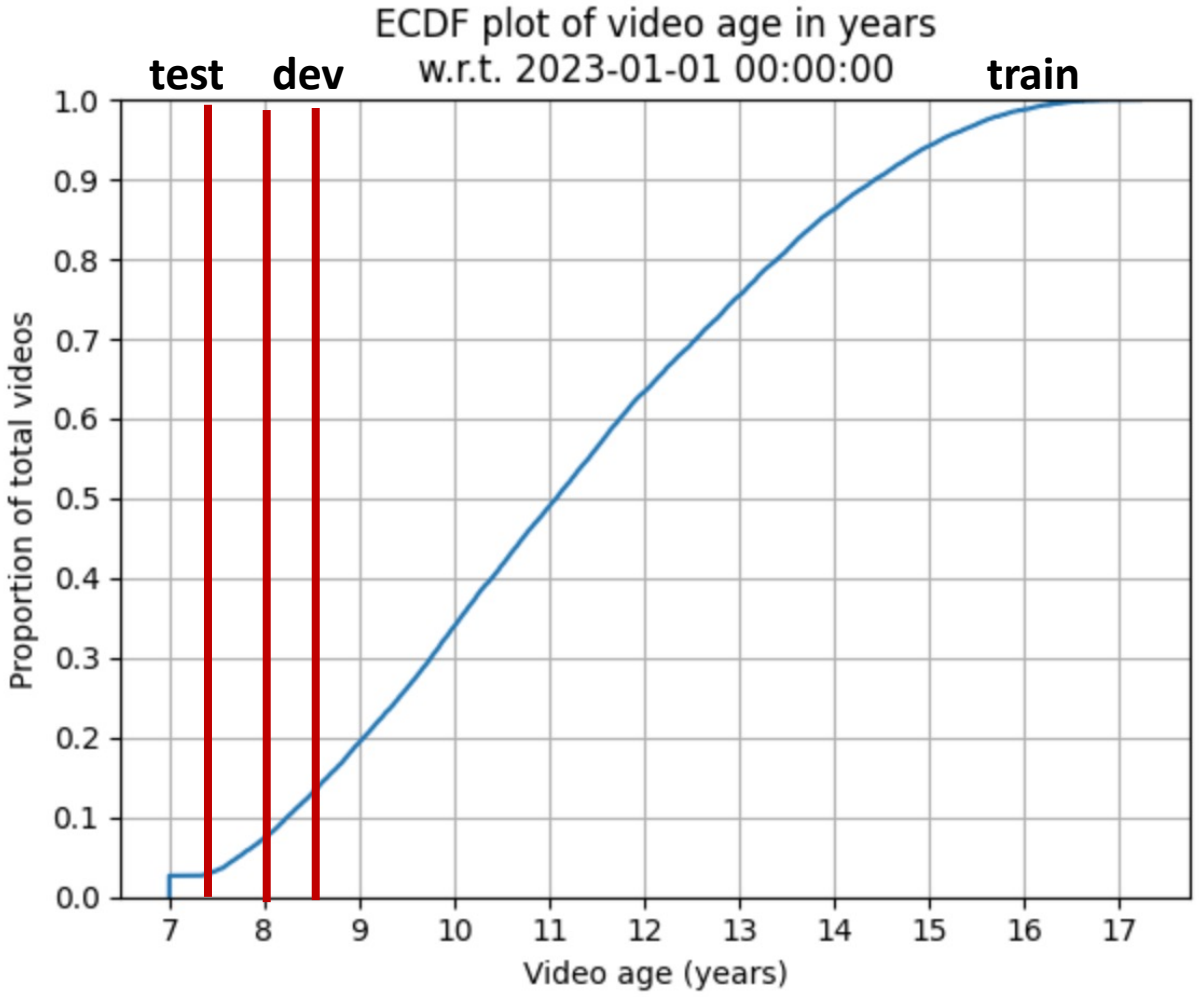


Minimum # comments: 5

Comment Count Stats

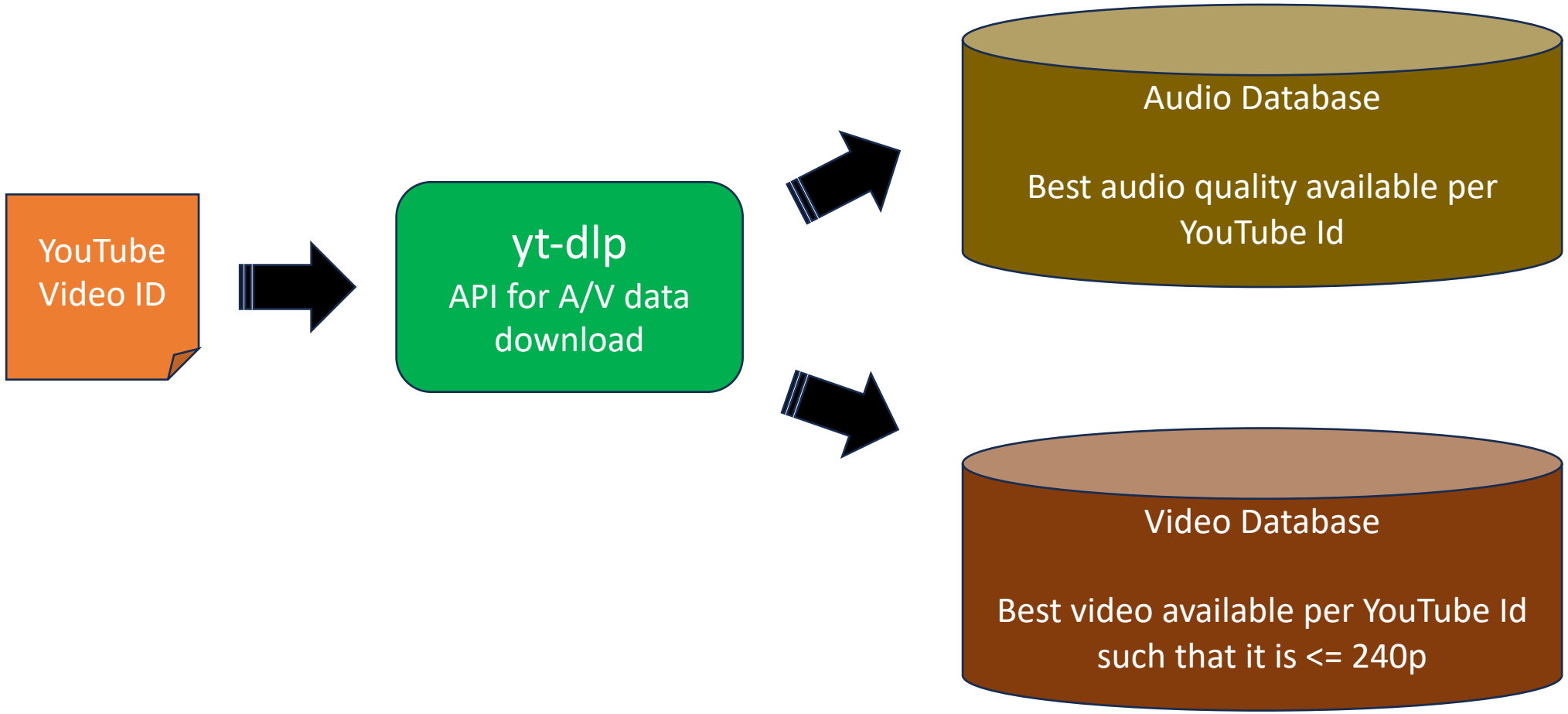
Mean: 236
Median: 34

train / dev / test splits: Based on Video Age



Test	7 - 8 years 527 ids
Dev	8 - 8.5 years 543 ids
Train	> 8.5 years 5087 ids

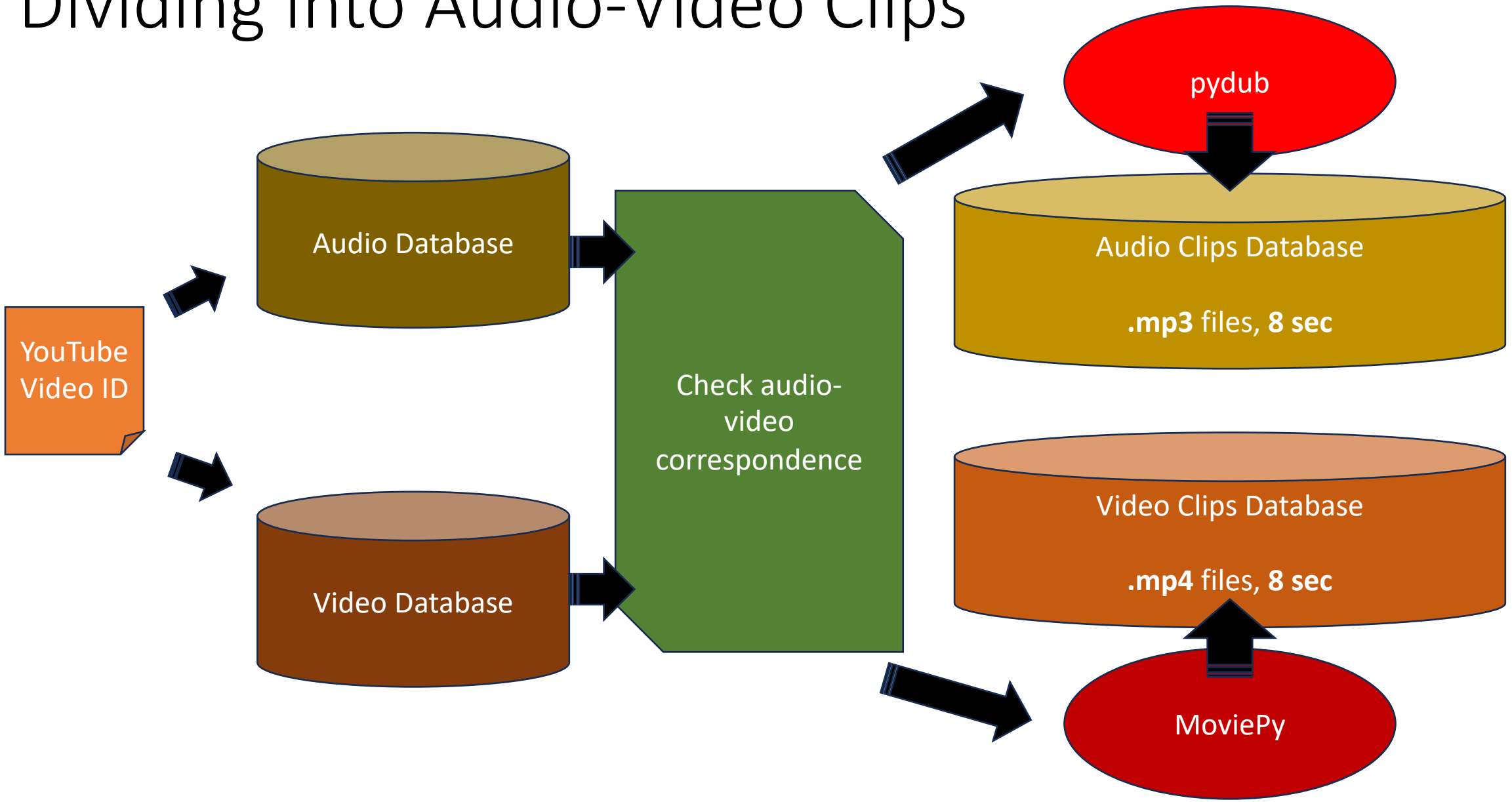
Audio & Video Download



Need for Further Preprocessing

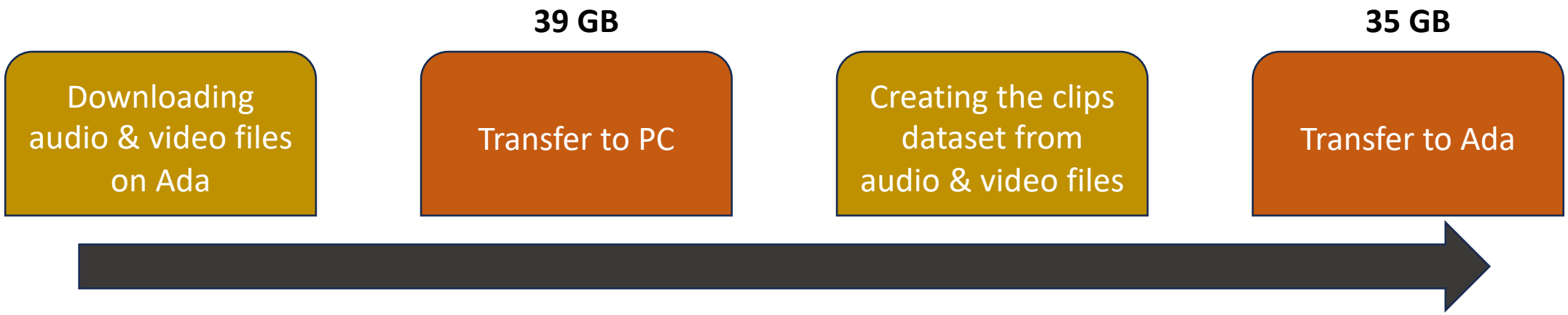
- Although videos are within 512 sec, the duration is still too high for current models for video processing!
- Audio and video files had different file formats
- Small number of YouTube Ids didn’t have corresponding pairs of audios and videos
- Video-wise dataset was small for model pretraining purpose, for which we just need small **“clips”** of corresponding audio-video pairs

Dividing Into Audio-Video Clips



Dividing Into Audio-Video Clips: Logistics

- Using the pydub and MoviePy libraries needed FFmpeg installation at the backend, not possible on Ada (compute server)
- Hence, the downloaded audios & videos were transferred back & forth from the local machine
- Significant overload in data transfer



Introducing Sparsity in Training Clips

- Creating the clips dataset from the complete training dataset would be data-heavy (especially on the video side)
- Hence, we sample clips out of different proportions of the training set with different levels of sparsity
- While sampling, we **randomly select x% of clips** from the total clips the video is divided into for saving to disk
- Dev and test sets are kept as-is, with all the clips taken
- Sampling training dataset with few-but-all clips, while keeping the evaluation dataset unharmed is widely used in literature

Training Clips Dataset: Sparsity Stats

% of training data	% clips sampled per video
30	20
20	25
20	33
20	50
5	75
5	100
Overall	**(approx.) 36%**

- On video-level problems, such technique can help make models more robust
- Models will be exposed to different videos with different no. of missing clips
- Hence, adaptable to incomplete data

Final Dataset Statistics

split	# clips	# videos
train	58,847	4,831
dev	15,961	499
test	14,806	461

mean audio/video duration	256.44 sec
mean # clips	31.55
mean # labels per video	2.30
mean # tags per video	13.10
mean # sampled clips per video (only applicable for train set)	12.18

audio dataset size	11 GB
video dataset size	24 GB

Agenda

- Introduction
- Literature Study
- Dataset Preparation
- Model Architecture
- Pretraining Objectives
- Training & Evaluation
- Concluding Remarks

Model Architecture

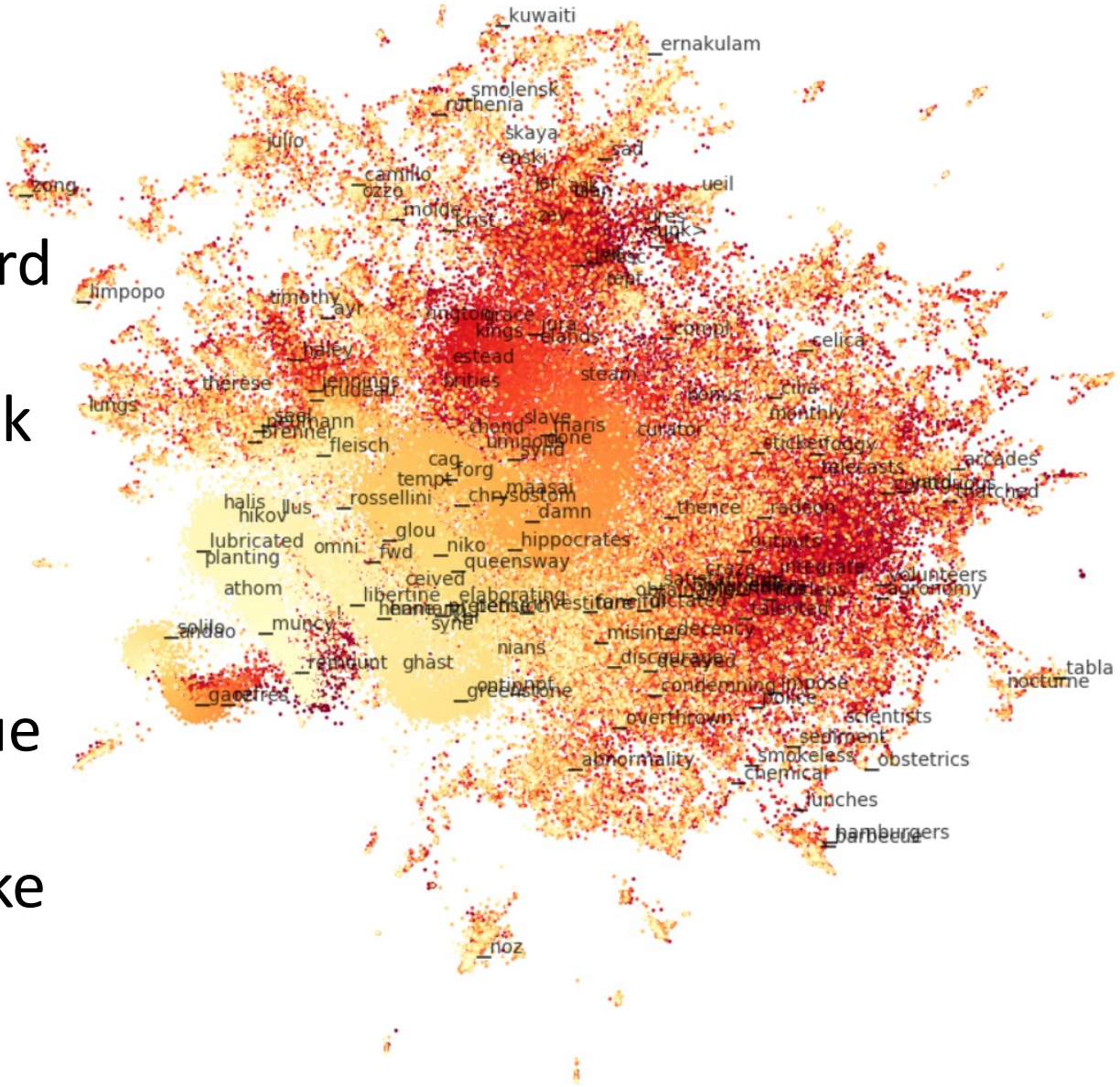
- A separate encoder for each modality
- Encoders are adapted to the music domain with our dataset
- Trained to project the representation for each modality in a common n-dimensional space

Text Modality

- We make use of the video labels and video tags as the associated text modality per clip
- For each clip, we randomly select one of the labels or clips as the text modality for that clip
- Issues with using other inputs for text:
  - Large diversity in videos, not all have lyrical content
  - Description or title often not found to be indicative enough
  - Many videos were non-English, hence problem would be multimodal+multilingual
  - Noisy alignment of already noisy text with audio/video modalities, needs careful preprocessing for quality alignment

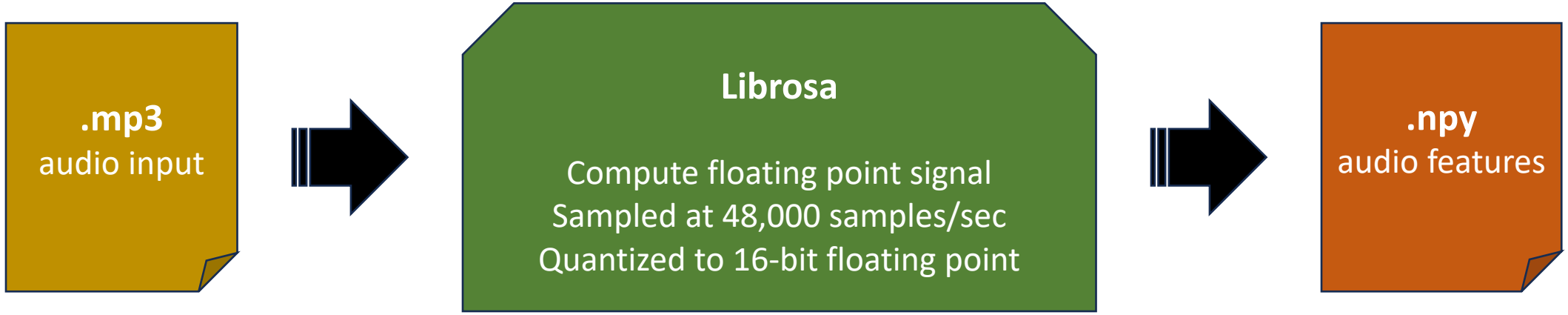
Text Encoder

- We make use of BPEmb for subword embeddings in English
- Build over a vocabulary size of 100k byte-pair tokens
- Provides a 300-dimensional representation
- Allows out-of-vocabulary words due to subword tokenization
- Suitable for bag-of-words inputs like in our case, where deep interpretation is not necessary



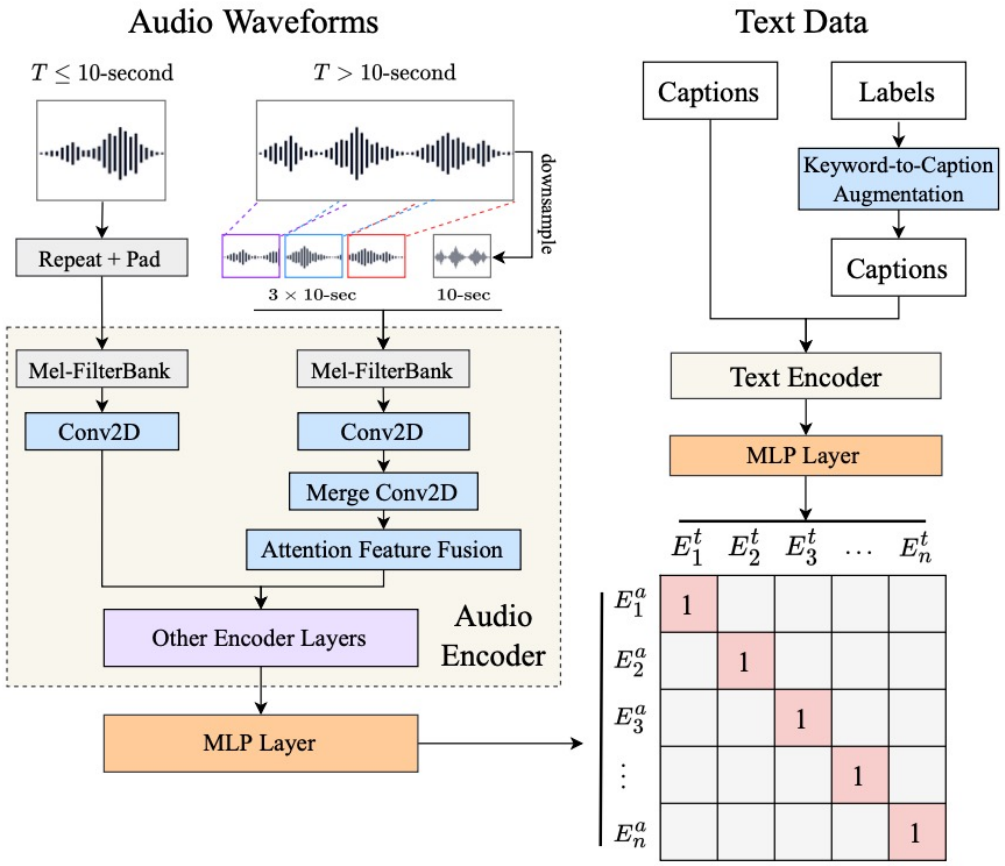
Audio Modality

- We use the prepared .mp3 audio clips files as the input per clip
- The files were used to precompute features with Librosa, and the outputs were serialized to individual .npy files prior to training
- Saves about **20 ms** of audio file processing (per file!) during training



Audio Encoder

- We make use of a pretrained CLAP (Contrastive Audio-Language Pretraining) model
- It consists of audio and text channels, from which we only use the audio channel
- The output of the audio channel is projected to a 300-dimensional space
- We use a variant of the audio encoder that is pretrained on music-related data in addition to general audio data

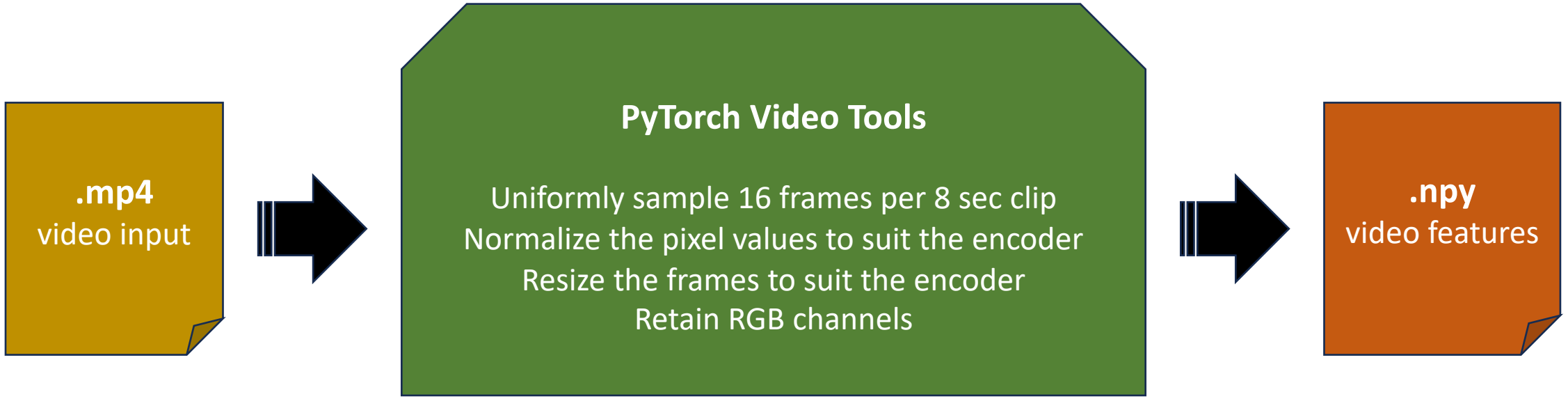


On CLAP

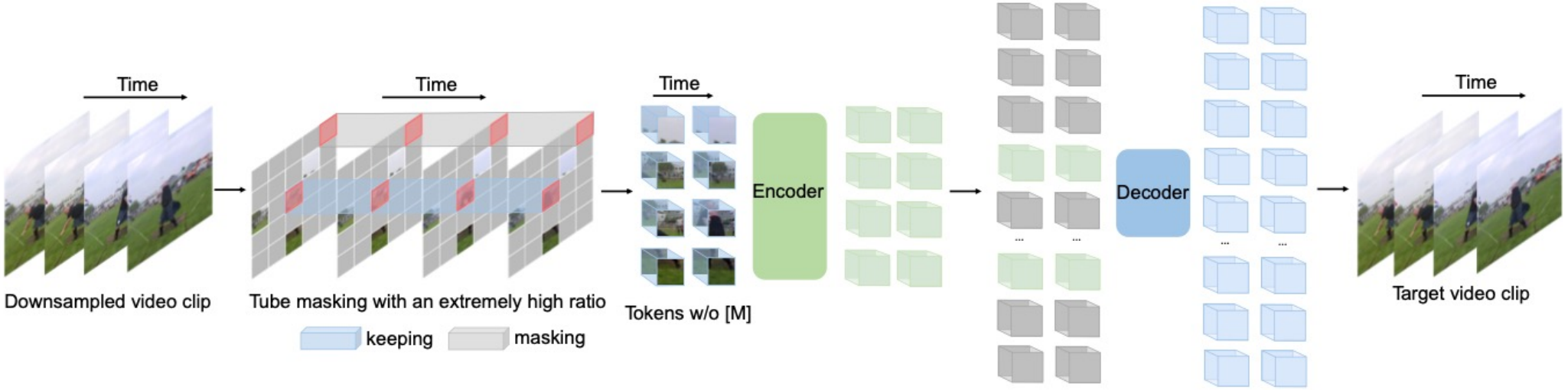
- The architecture uses contrastive pretraining on a 634k-size dataset of audio-caption pairs
- Efficacy of the pretrained model was shown on these downstream tasks:
  - Zero-shot text-to-audio retrieval
  - Zero-shot audio classification
  - Supervised audio classification
- For pretraining, tags available with the audio files were utilized to construct corresponding captions as:
  - The sound of `label-x`
- Contrastive training objective was same as the popular CLIP architecture for image-text pretraining

Video Modality

- We use the prepared .mp4 video clip files as the input per clip
- The files were used to sample frames and serialize their pixel values into .npy files prior to training
- Saves about **200 ms** of video file processing (per file!) during training

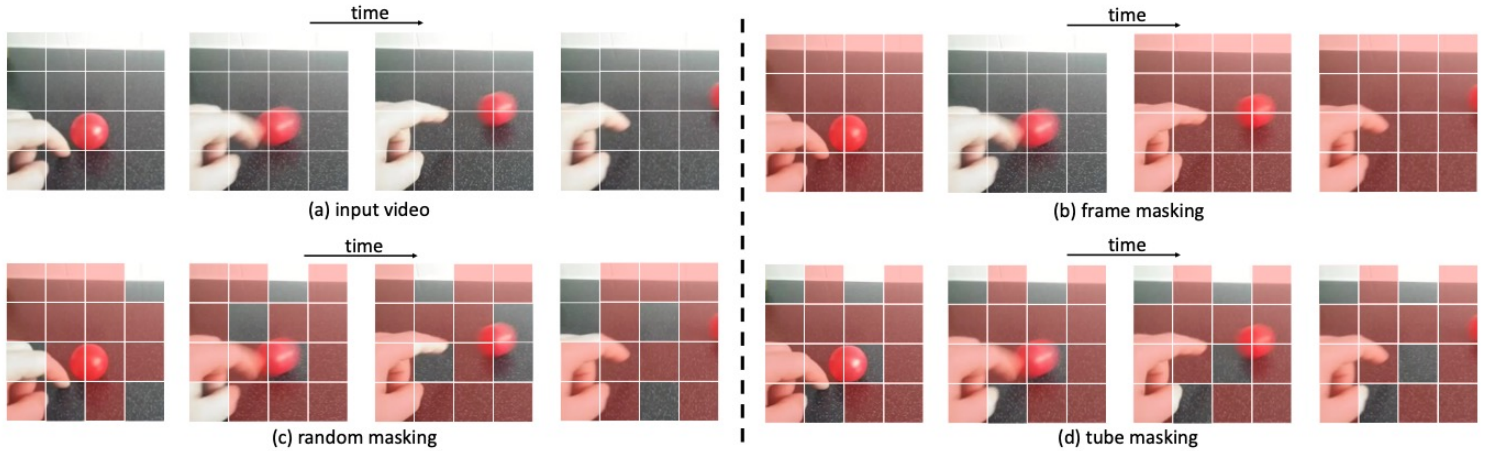


Video Encoder



- VideoMAE is used as the video encoder channel, the output of which is projected to a 300-dimensional space
- The transformer-based model is unimodal to provide a strong representation for videos
- Backbone consists of joint space-time attention for video signal processing

More on VideoMAE



- The model is self-supervised pretrained on a **M**asked **A**uto**E**ncoder objective, where masked input is to be reconstructed at the output
- The work proposed a tube masking strategy for masking of video inputs
- Masking ratio is kept high (90-95%) due to temporal redundancy in videos

Model Stats

Channel	# parameters
text channel	3.1 M
audio channel	74 M
video channel	86 M
Model	**163 M**

The model occupies a size of 626 MB on disk.

Agenda

- Introduction
- Literature Study
- Dataset Preparation
- Model Architecture
- Pretraining Objectives
- Training & Evaluation
- Concluding Remarks

On Model Training

- Our objective is to adapt the model to our domain of music associated with the 45 identified labels
- To adapt the model to our domain, we train the model to learn to associate the related clips pairs of modalities closer:
  - text-audio
  - text-video
  - audio-video
- At the same time, push unrelated pairs (from different videos / clips) farther from each other in the common space
- We experiment with different pretraining objectives for this purpose

Training Instance

Each training instance comprises of the three modalities associated with each other, for each clip

Associated text for the clip
Associated audio for the clip
Associated video for the clip

Pretraining Objectives

- Two types of pretraining strategies have been tried out for adapting our developed architecture:
  1. Cosine Similarity
  2. Contrastive Learning
- Each involve 3 types of interactions between modality representations:
  - text-audio
  - text-video
  - audio-video
- We also try weighted versions of both the pretraining objectives

Cosine Similarity

$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

- Two types of sample pairs:
  - Positive pairs: from the same training instance
    To tech the model what types of input pairs are to be represented similarly
  - Negative pairs: from different training instances
    To teach the model what types of input pairs are to be represented differently
- For negative pairs, we randomly sample instances from the same training batch

Trimodal Cosine Similarity: Terms

- Consider a positive instance:
  (T_p, A_p, V_p)
- Randomly sampled negative instance:
  (T_n, A_n, V_n)
- Positive term expression for cosine objective component:
  P-COS(A, B) = 1 - cos(A, B)
  Farther the embeddings, the higher the loss
- Negative term expression for cosine objective component:
  N-COS(A, B) = max(0, cos(A, B) - margin)
  Closer the embeddings, the higher the loss

Trimodal Cosine Similarity Objective

The unweighted variant of the trimodal cosine similarity objective can be given as below:

P-COS(Tp, Ap) + P-COS(Tp, Vp) + P-COS(Ap, Vp) +
N-COS (Tp, An) + N-COS(Tp, Vn) + N-COS(Ap, An)

Since we could leverage a batch size of only 2, the other instance in the batch per given instance is selected as the negative instance in this computation

Weighted Cosine Similarity Objective

The weighted variant of the trimodal cosine similarity objective can be given as below:

0.15*P-COS(Tp, Ap) + 0.15*P-COS(Tp, Vp) + 0.7*P-COS(Ap, Vp) + 0.15*N-COS (Tp, An) + 0.15*N-COS(Tp, Vn) + 0.7*N-COS(Ap, An)

We apply a high weight to the audio-video pairs since they are much stronger signals backed by much more deeper architectures, as compared to text signals, which are noisy for many clips, also backed by simplistic encoder channel

Contrastive Learning

- The contrastive learning objective learns this task:
  Given a batch of **B** (**a**, **b**) pairings, we have **B** * **B** possible pairings for **a** & **b**
  Out of all these possible pairings,
  Predict which of the **B** pairings are true (i.e. actually occur in the batch)
- Example formulation for the audio-text modality pair: (from the CLAP paper)

$$L = \frac{1}{2N}\sum_{i=1}^{N}(\log\frac{\exp(E_i^a \cdot E_i^t/\tau)}{\sum_{j=1}^{N}\exp(E_i^a \cdot E_j^t/\tau)} + \log\frac{\exp(E_i^t \cdot E_i^a/\tau)}{\sum_{j=1}^{N}\exp(E_i^t \cdot E_j^a/\tau)})$$

Implementing Contrastive Learning Objective

Sample computation for audio-video modality:

Training batch: (T_B, A_B, V_B), batch size = S

$E_A = \text{AudioChannel}(A_B)$

$E_V = \text{VideoChannel}(V_B)$

$P = E_A \cdot E_V$

$y = [1, 2, \ldots, S]$

$L_A = \text{CrossEntropy}(P, y)$

$L_V = \text{CrossEntropy}(P^T, y)$

$L_{AV} = \frac{1}{2}(L_A + L_V)$

The matrix P for S = 3

$A_1.V_1$	$A_2.V_2$	$A_3.V_3$
$A_2.V_1$	**$A_2.V_2$**	$A_2.V_3$
$A_3.V_1$	$A_3.V_2$	**$A_3.V_3$**

Illustration of the cross-entropy loss

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \quad CE = -\sum_i^C t_i log(f(s)_i)$$

Trimodal Contrastive Learning Objective

The unweighted contrastive learning objective can be given as the simple sum of the terms for the objectives computed for the individual modality pairs:

$L_{TAV} = L_{TA} + L_{TV} + L_{AV}$

The weighted version, to focus on audio-video modality matching:

$L_{TAV} = \mathbf{0.15}*L_{TA} + \mathbf{0.15}*L_{TV} + \mathbf{0.7}*L_{AV}$

However, since the batch size was 2, every pair was contrasted against the only other pair from the batch

Agenda

- Introduction
- Literature Study
- Dataset Preparation
- Model Architecture
- Pretraining Objectives
- Training & Evaluation
- Concluding Remarks

Pretraining Settings / Hyperparameters

# epochs	5
batch size	2
accumulate steps	4
effective train batch size	8
peak learning rate	6e-5
learning rate schedule	warmup till 10% training followed by linear decay
optimizer	AdamW
validation frequency	2 / epoch
best checkpoint criteria	minimum average distance between positive audio-video pairs in dev set

Pretraining: Results

Values on best checkpoint on three metrics calculated on the dev set.

The distances are computed as the average of cosine similarity between all the positive cross-modal pairs in the evaluation dataset.

Experiment	Audio-Video Distance	Text-Audio Distance	Text-Video Distance
Cosine similarity	0.44	**0.84**	**0.84**
Weighted cosine similarity	**0.41**	0.91	0.90
Contrastive	0.68	0.95	0.96
Weighted contrastive	0.80	0.98	0.97

Cosine similarity seems to be a more suitable learning objective, based on these numbers.

Additional Evaluation of Adapted Models

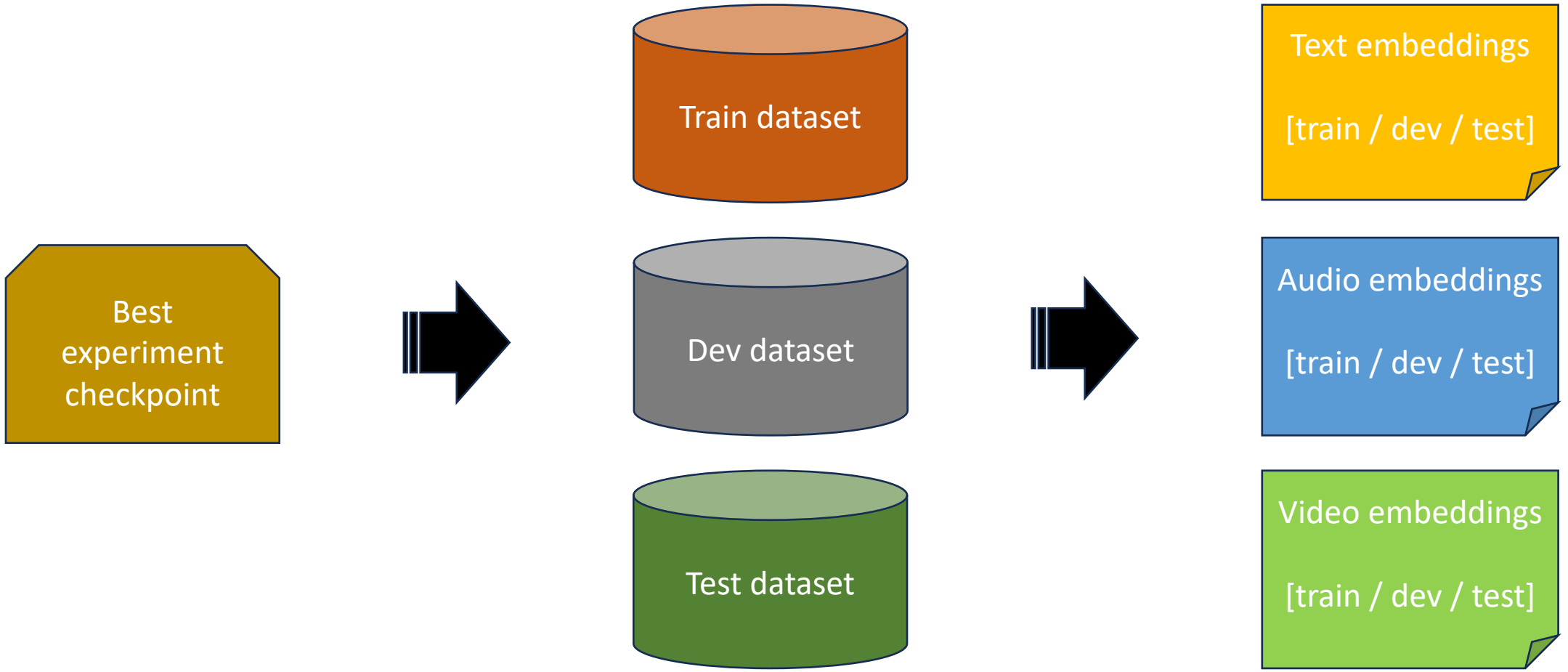
We make use of these 2 additional tasks to check the efficacy of the embeddings from our pretrained models:

1. Classification from audio-video signals
   Involves video-level classification into one or more 45 identified categories using audio & video features.
2. Cross-modal Retrieval
   Retrieving one modality by using the other as the input.

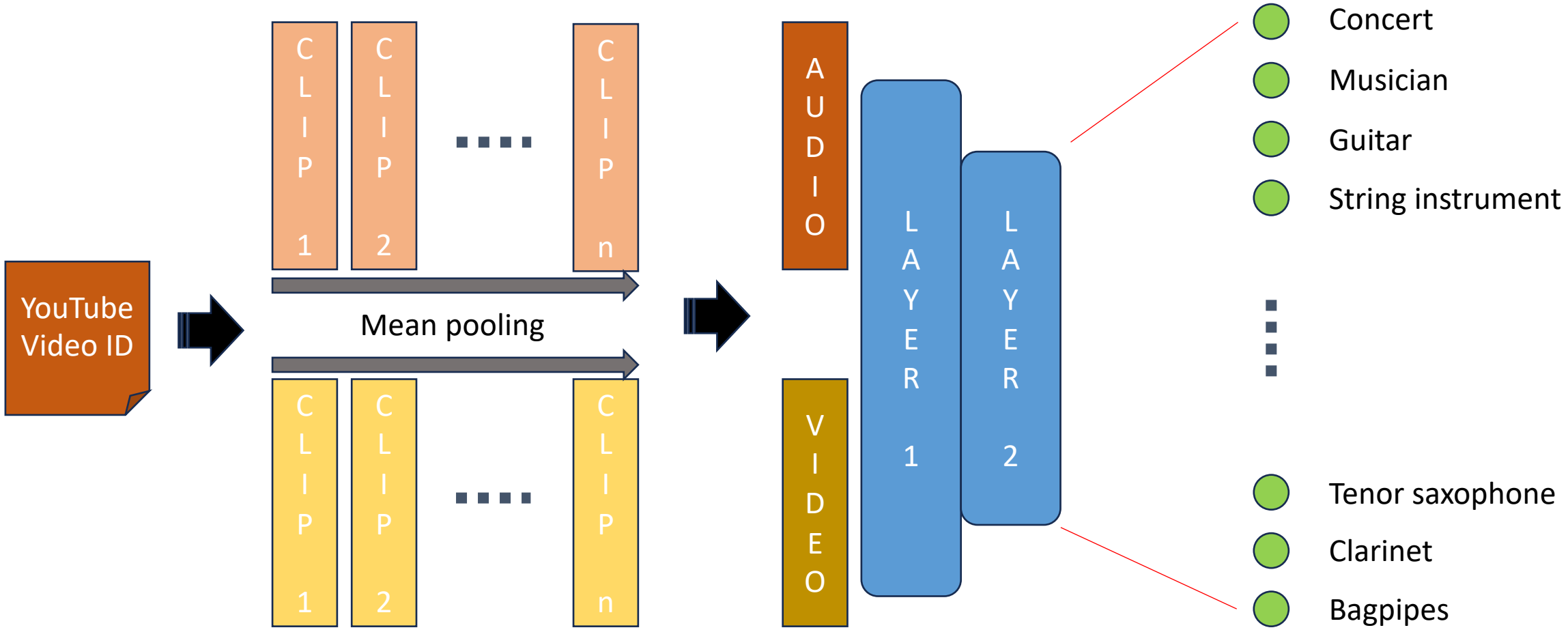
It does not involve any further tuning of the model parameters.

The best model checkpoint per experiment is only used to obtain the embeddings, post which a small task-specific model is trained.

Serialization of Trained Embeddings



Modeling for Multilabel Video Classification

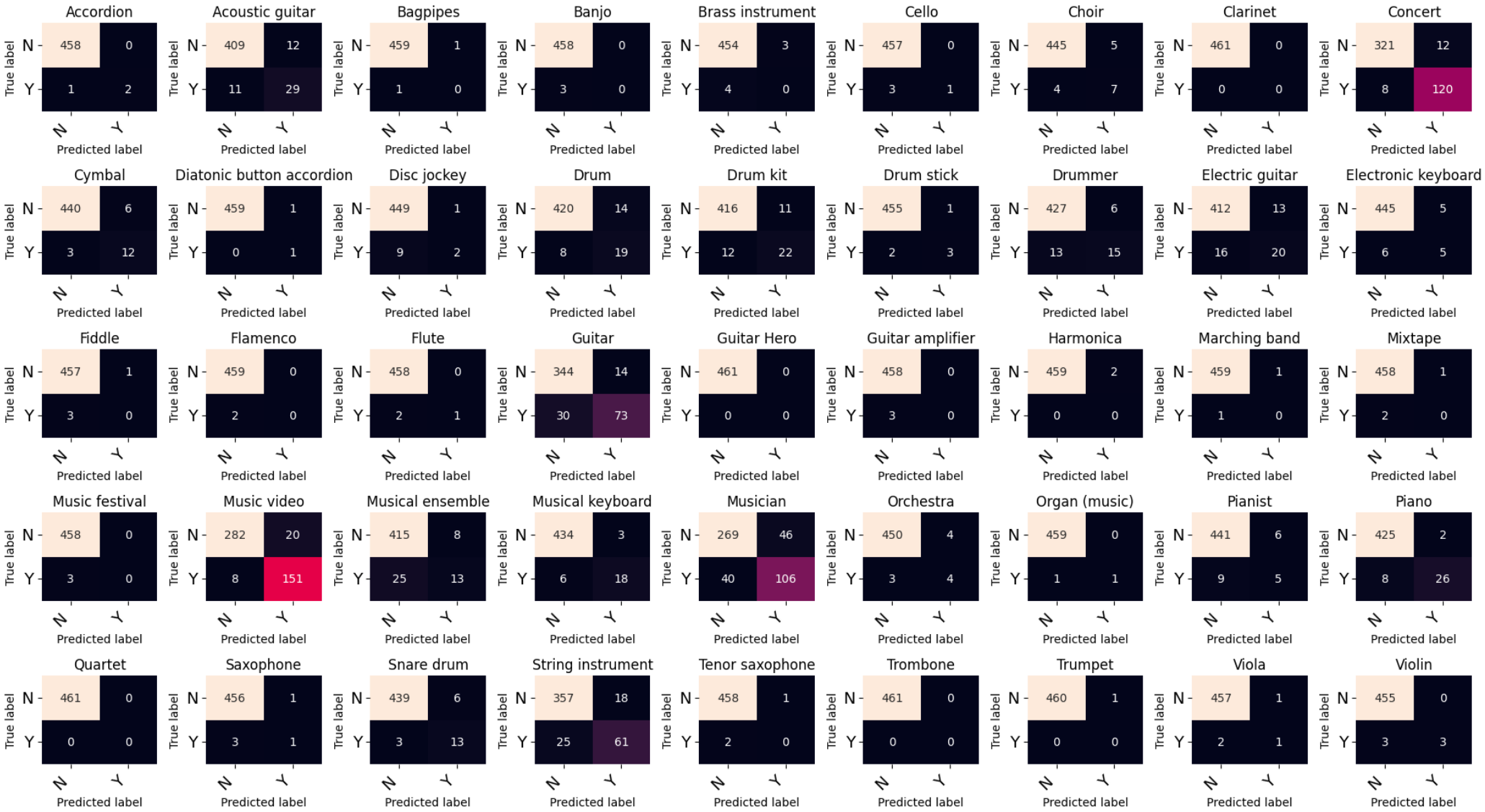


Multilabel Video Classification: Results

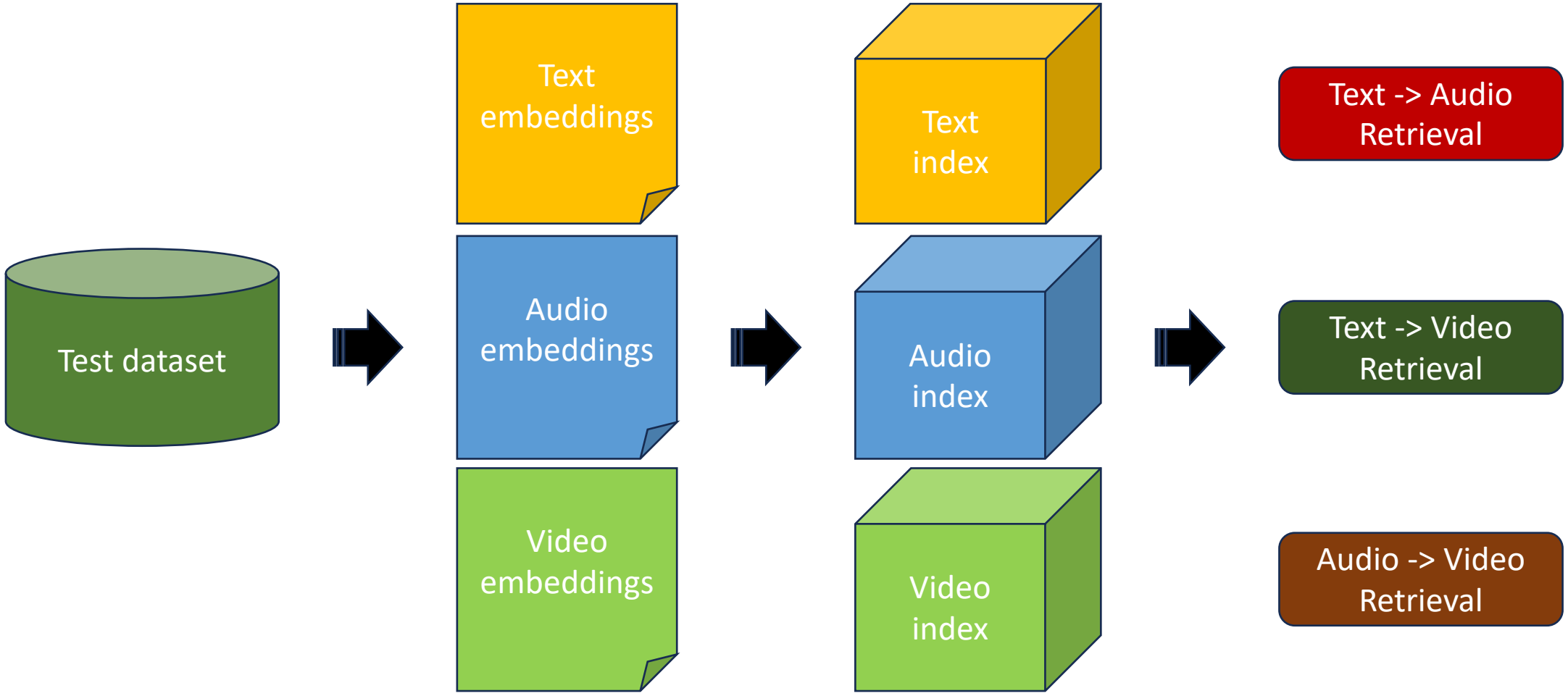
Experiment	F1	Precision	Recall	Accuracy	Hamming Loss
Cosine similarity	64.26	68.53	61.88	**62.28**	0.0324
Weighted cosine similarity	55.55	65.71	50.93	60.28	0.0350
Contrastive	71.03	73.32	70.38	45.56	0.0265
Weighted contrastive	**73.50**	**77.24**	**71.46**	49.02	**0.0233**

The representations obtained from the weighted contrastive experiment give the best results, in contradiction to the results obtained in distance-based comparison.

Confusion Matrix for the Weighted Contrastive Model



Trimodal Retrieval from Trained Embeddings



Text-to-Audio Retrieval: Results

Experiment	Recall @ 1	Recall @ 5	Recall @ 10	Mean Rank	Median Rank
Cosine similarity	1.30	5.42	9.98	117.62	83.0
Weighted cosine similarity	0.65	4.12	8.03	121.46	81.0
Contrastive	**1.08**	6.29	11.71	106.72	77.0
Weighted contrastive	**1.08**	**7.59**	**13.23**	**106.56**	**73.0**

The “Weighted contrastive” model performs the best, consistently on all the calculated retrieval metrics. On the other hand, the ”Weighted cosine similarity model” performs the worst.

Text-to-Video Retrieval: Results

Experiment	Recall @ 1	Recall @ 5	Recall @ 10	Mean Rank	Median Rank
Cosine similarity	0.65	2.82	5.86	138.12	98.0
Weighted cosine similarity	0.65	3.25	6.07	122.91	87.0
Contrastive	0.87	**5.21**	10.20	**113.20**	**82.0**
Weighted contrastive	**1.08**	4.56	**11.28**	114.97	83.0

The performance of the weighted and non-weighted contrastive models are competitive across metrics, with the non-weighted version having a slight edge. Both the cosine similarity models perform worse as compared to contrastive models.

Audio-to-Video Retrieval: Results

Experiment	Recall @ 1	Recall @ 5	Recall @ 10	Mean Rank	Median Rank
Cosine similarity	0.65	3.90	8.46	107.17	80.0
Weighted cosine similarity	0.87	3.69	7.59	107.94	77.0
Contrastive	0.43	3.90	9.33	108.66	83.0
Weighted contrastive	**2.39**	**6.07**	**12.58**	**90.18**	**60.0**

The weighted contrastive model performs exceedingly well across all the metrics. Both the weighted versions outperform their non-weighted counterparts.

Overall Analysis

- The results from distance-based metrics versus multilabel classification & cross-modal retrieval show exactly opposite trends: this indicates mere distance between positive pairs not indicative of the embedding performance.
- The contrastive learning objective was applied on batch sizes as high as 32,768 on large, distributed GPUs. In our case, we had a batch size of only 2, hence limited in achievable performance.
- Study of convergence curves while model training pointed out the need for experiment-specific hyperparameter tuning.

Agenda

- Introduction
- Literature Study
- Dataset Preparation
- Model Architecture
- Pretraining Objectives
- Training & Evaluation
- Concluding Remarks

Summary: 1

- We started off with a study of different works in literature with respect to music videos in specific, and found lack of computational methods applied for such studies.
- Further, it was realized that for the few works that study multimodal representation of music, most are bimodal in nature: i.e. do not involve a singular architecture to understand the textual, audio and video components in music at once.
- Hence, we first worked to build a dataset that can be used to apply trimodal representational learning approaches. This was curated by identifying 45 music-related categories from YouTube 8M dataset.

Summary: 2

- We came up with a model architecture consisting of 3 channels for individual modality processing. Each were pretrained on generic data, and were to be adapted to the music domain into a common representational space for all the modalities.
- We designed two types of pretraining objectives, and implemented model training using their weighted and un-weighted versions.
- We evaluated our pretraining experiments using cross-modal retrieval and downstream training on the multilabel video-level classification problem.

Future Work: Short Term

- Computational optimization of the training process (we currently have a batch size of only 2)
- Training for more number of epochs
- Hyperparameter optimization
- Trying more pretraining objectives such as the InfoNCE loss
- Downstream evaluation on more tasks, such as audio classification on GTZAN, MagnaTagATune datasets
- Experimenting with different pretrained models for modality representation

Future Work: Long Term

- Improving the text modality representation to include more sophisticated text signals such as lyrics
- Focusing on more specific domains within music such as music videos
- Enhancing understanding from the music-adapted models with:
  - Identification of themes in music videos
  - Demographic representation in videos
  - Correlation of audio, video and textual characteristics:
    - for a given set of lyrics, what type of video is expected?
    - given a video clip, what are the most suitable set of lyrics & music?
- Experimenting with generational objectives on top of the trained representations

Thank you!

Code:
github.com/sagarsj42/trimodal-music-representation

Contact: sagar.joshi@research.iiit.ac.in