VMCML: Video and Music Matching via Cross-Modality Lifting

Yi-Shan Lee[1] Wei-Cheng Tseng[2,3] Fu-En Wang[1] Min Sun[1]
[1]National Tsing Hua University [2]University of Toronto [3]Vector Institute
yishanlee@m110.nthu.edu.tw weicheng.tseng@mail.utoronto.ca
fulton84717@gapp.nthu.edu.tw sunmin@ee.nthu.edu.tw

Abstract

*We propose a content-based system for matching video and background music. The system aims to address the challenges in music recommendation for new users or new music give short-form videos. To this end, we propose a cross-modal framework VMCML (Video and Music Matching via Cross-Modality Lifting) that finds a shared embedding space between video and music representations. To ensure the embedding space can be effectively shared by both representations, we leverage CosFace loss based on margin-based cosine similarity loss. Furthermore, to confirm the music is not the original sound of the video and that more than one video is matched to the same music, we follow the rule and collect videos and music from a well-known multi-media platform. That is because there are limitations of previous datasets. We establish a large-scale dataset called **MSV**, which provide 390 individual music and the corresponding matched 150,000 videos. We conduct extensive experiments on **Youtube-8M** and our **MSV** datasets. Our quantitative and qualitative results demonstrate the effectiveness of our proposed framework and achieve state-of-the-art video and music matching performance.*

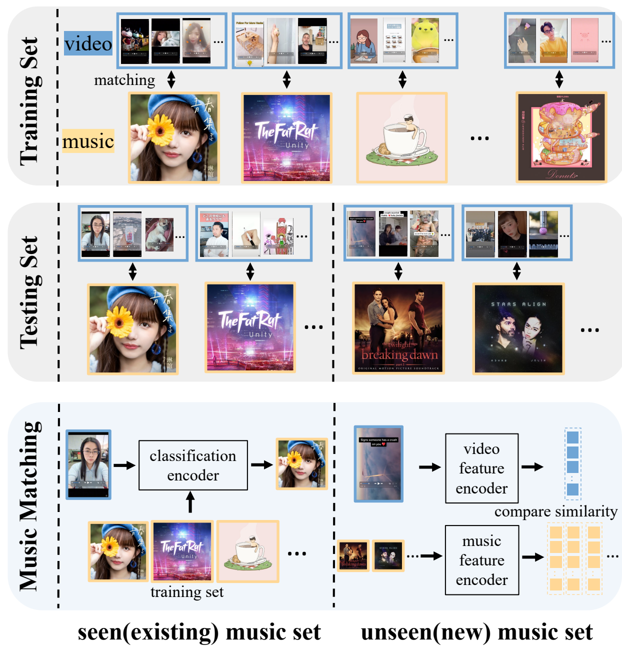

Figure 1. **The content-based video and music matching system.** For training, we collect several pieces of music and several videos using the same background music. Hence, one music matches several videos. In testing, seen music set (existing music in training) can be addressed as classifying a video to a set of music identities. The unseen music set (new music unseen in training) can be viewed as performing video verification by comparing the similarity between the video and the new music identities.

1. Introduction

In recent years, short-form videos have rapidly entered our daily lives. People record their life by uploading short videos to various platforms such as TikTok, Instagram Reels, and Youtube Shorts. The daily usage time of TikTok by young generation users (ages 10-25) in 2022 has grown by 2.38 times from 2017 to 2022, with 44 minutes into 105 minutes of daily TikTok. After the opening of Instagram Reels in April 2022, the number of short-form videos has increased significantly with an increase of 971 percent in August compared to April. These platforms typically provide context-based recommendation systems to help users attach background music to their uploaded videos. These music are mostly selected according to users' previous selection or current trends, which eventually bias towards a few existing trendy background music. The diversity of background music is important for these platforms since new trendy music can be more engaging and the uniqueness of trendy music can attract more new users to the platform. However, the contents of background music and videos are typically not considered.

The content-based music matching system, as shown in Figure 1, is crucial for (1) recommending new music as they haven't been selected and (2) recommending existing music to new users as they lack of previously selected music. There are few pioneers who work on cross-modal matching for video and audio content, while previous works have focused on cross-modal matching for images and text. The pairing of visual and language elements usually follows a one-to-one mapping. However, when it comes to visual and audio, there are instances where the pairing becomes one-to-many. This makes the cross-modal matching for video and audio more challenging. Hong *et al*. [9] proposed a content-based retrieval model that combines the inter-modal ranking loss and soft intra-modal structure loss to construct a shared embedding space. Surís *et al*. [21] (referred to as CEVAR) proposed a joint embedding model with the classification loss along with a similarity loss, which incorporates the video labels provided by the YouTube-8M [1] dataset. Yi *et al*. [29] (referred to as CMVAE) proposed a hierarchical Bayesian generative model using variational autoencoder and matching relevant background music to videos by the corresponding latent embeddings. However, all these methods exhibit a significant performance gap when matching existing music to new videos compared to matching new music to new videos. We use seen or existing interchangeably and unseen or new interchangeably.

We treat this cross-modal matching task as a metric-learning problem where most music matches to a few videos as they are non-trendy. Moreover, we want our method improves on both recommending existing music, as well as new music. It is not obvious at first glance, but face recognition [12,17,20,23] shares the same challenges to our task. A face recognition model not only needs to recognize existing faces in training, but it also needs to enroll new faces and identify them without retraining the model. In addition, the model should only require a person to enroll a few faces for ease of usage. Inspired by these observations, we adopt the CosFace loss [24] and the ArcFace loss [4], widely used for face recognition to our cross-modal matching task. We refer to these losses as lifting loss. To find a shared embedding space between video and music, there is a shared head that makes both video and music features more aligned. Furthermore, we calculate the similarity loss between video and music features. More specifically, we combine cross-modality lifting loss and cross-modality similarity loss for cross-modal matching.

The public datasets directly downloadable for matching video and background music features are limited. We established a dataset, Music for Short Video Dataset **MSV**, containing 150k videos and 390 corresponding background music pairs. The differences between Youtube-8M [1] and MSVD are (1) the ground truth of music and video in the former is a one-to-one mapping but there is more than one video matched to the same music in the latter, (2) most of the music in the former is the original of the video and (3) most of the videos in the former are longer than 1 minute. The distinction between Tiktok dataset [10] and **MSV** lies in their content. The Tiktok dataset [10] exclusively comprises videos, whereas MSV includes both the video and their accompanying music. We divide the MSVD into seen music set and unseen music set corresponding to real-world scenarios. With the **MSV** dataset established, we conduct experiments to evaluate the performance of VMCML and compare it with previous methods.

Our main contributions are summarized as follows:

(1) We propose a novel cross-modal framework VMCML with cross-modality lifting loss and cross-modality similarity loss for content-based background music matching, which can be applied to both seen and unseen video-music samples.

(2) We collect a short video and background music matching dataset called "Music for Short Video (**MSV**)", the suitable dataset for the video-music matching task which provides 390 music and their 150,000 corresponding videos.

(3) VMCML achieve the state-of-the-art on **MSV** and Youtube-8M [1].

2. Related Works

We introduce the previous cross-modal matching methods in Sec. 2.1, and present the metric learning in Sec. 2.2.

2.1. Cross-Modal Matching

Most existing cross-modal matching methods typically focus on textual and visual modalities, using open-sourced datasets such as MSCOCO [16], Flickr30k [19], ActivityNet-captions [14] and MSR-VTT [28]. Zhen *et al*. [31] proposed a deep supervised cross-modal retrieval method that simultaneously minimizes discrimination loss in the label space and invariance loss. Wei *et al*. [26] introduced a universal weighting metric learning framework and a new polynomial loss under it. The universal weighting framework provides a powerful tool to analyze various metric-learning-based weighting and loss, which have been widely used in cross-modal matching. However, there has been little effort devoted to matching video and music content. There are few differences between visual-language learning and visual-audio learning when it comes to cross-modality matching. The pairing of visual and language elements typically follows a one-to-one mapping. However, in the case of visual and audio, there are instances where the pairing becomes one-to-many. This situation makes the task more challenging. Previous work commonly used cross-modal ranking losses for music and video [9, 15, 30]. The TikTok Dataset [10] offers videos along with human mask and human UV coordinates but no music part. Hong *et al*. [9] presented a content-based retrieval model that only uses content features between music and videos, constraining the relative distance relationship of samples within each modality. While Suris *et al.* [21] uses the visual features and audio features provided by Youtube-8M [1] to minimize the distance of visual embedding and audio embedding of the same video at representation space to predict the corresponding video label. Yi *et al.* [29] proposed a hierarchical Bayesian generation model using the variational auto-encoder to match relevant background music to video through their latent embedding. Nonetheless, these methods face a significant performance gap when matching existing music to new videos compared to matching new music to new videos. This work introduces face recognition-inspired loss to mitigate the gap.

2.2. Metric learning

The goal of metric learning is to learn the similarity between features to enable accurate feature matching and verification. To improve the quality of feature embedding, contrastive loss [2, 3] and triplet loss [8, 25] are commonly employed techniques that help increase the Euclidean margin. More specifically, the former ensures that same-class objects are mapped together in the representation space, while different-class objects are mapped apart by a margin. The latter compares distances between anchor, positive and negative samples. In addition, there are other variations of metric learning methods such as center loss [27] and angular loss [24] that have been proposed specifically for face recognition tasks. Center loss aims to minimize the distance between each sample and its corresponding class center. In contrast, angular loss minimizes the distance between the feature embedding and its corresponding class boundary. Overall, metric learning has played a crucial role in enabling accurate verification systems for learning the similarity between features.

3. Approach

In this section, we introduce the problem formulation of the video-music matching task in Sec. 3.1. Then, we detail our loss functions in Sec. 3.2 for lifting cross-modality features to a shared space. Finally, we introduce our proposed VMCML framework for video-music matching in Sec. 3.3.

3.1. Problem Definition

Given a music set $\mathcal{M} = \{\mathbf{m}_i | i \in \mathbb{Z}, 1 \leq i \leq N_{\mathbf{m}}\}$ with $N_{\mathbf{m}}$ music and a video set $\mathcal{V} = \{\mathbf{v}_i | i \in \mathbb{Z}, 1 \leq i \leq N_{\mathbf{v}}\}$ with $N_{\mathbf{v}}$ videos from training dataset, where $(\mathbf{m}_i, \mathbf{v}_i)$ denote music features and video features. For matching video and music, we adopt the shared weight $\mathbf{W}$ as the prototype to lift video and music features to a shared embedding space. Our video-music matching task is formulated as a mapping $f : \mathbf{m}_i \to y_i$ and $f : \mathbf{v}_i \to y_i$, where y_i denotes the predicted matching music class labels in the training dataset. During the testing stage, video and music can be matched by estimating the cosine similarity between their features, i.e., $\cos(\mathbf{m}_i, \mathbf{v}_i)$. For music already included in the training dataset, we can directly use y_i as the matched one.

3.2. Cross-Modality Training Objectives

Softmax loss is commonly used in the classification problem to minimize intra-class and maximize inter-class distances, which is formulated as:

$$L_S(\mathbf{x}_i, \mathbf{W}) = -\log \frac{e^{\mathbf{W}_{y_i} \cdot \mathbf{x}_i}}{\sum_{k=1}^{N} e^{\mathbf{W}_k \cdot \mathbf{x}_i}} , \tag{1}$$

where $\mathbf{W}$ is the prototype, i.e., the weight of the last layer in a network, $\mathbf{W}_k$ is the weight of the k-th class, N is the total number of classes, $\mathbf{x}_i$ is the feature, and y_i is the ground truth class label of $\mathbf{x}_i$. In this paper, the $\mathbf{x}_i$ is either video features or music features. To further improve the decision boundary between different classes, CosFace [24] proposed lifting the features and prototypes to a hyper-sphere by introducing a scaling term s and a margin μ:

$$\begin{aligned} \cos(\theta_{k,i}) &= \frac{\mathbf{W}_k \cdot \mathbf{x}_i}{\|\mathbf{W}_k\| \cdot \|\mathbf{x}_i\|} , \\ L_C(\mathbf{x}_i, \mathbf{W}) &= -\log \frac{e^{s \cdot [\cos(\theta_{y_i,i}) - \mu]}}{e^{s \cdot [\cos(\theta_{y_i,i}) - \mu]} + \sum_{k \neq y_i}^{N} e^{s \cdot \cos(\theta_{k,i})}} . \end{aligned} \tag{2}$$

Since Equation (2) is based on the angles between intra and inter classes (i.e., $\theta_{y_i,i}$ and $\theta_{k,i}$) in a normalized feature space, the features $\mathbf{x}_i$ are optimized within a hyper-sphere.

Cross-Modality Lifting Loss. To solve the video-music matching task as a metric learning problem, we aim at lifting video and music features to the same hyper-sphere. In this way, we can match the videos to their most appropriate music by calculating the cosine similarity between them. Hence, we propose the "Cross-Modality Lifting Loss" by adopting a shared prototype $\mathbf{W}$ for both video and music features, and consider the modality-to-prototype distances:

$$L_{LL}(\mathbf{v}_i, \mathbf{m}_i, \mathbf{W}) = L_C(\mathbf{v}_i, \mathbf{W}) + \alpha L_C(\mathbf{m}_i, \mathbf{W}) , \tag{3}$$

where $(\mathbf{v}_i, \mathbf{m}_i)$ are input music feature and video feature, and α is a hyper-parameter.

Cross-Modality Similarity Loss. Although our proposed cross-modality lifting loss can effectively minimize the intra and maximize the inter class distances, we found that only considering the modality-to-prototype distance still leads to a sub-optimal performance since videos and music are eventually matched based on their features instead of their prototypes during the testing stage. In order to overcome this

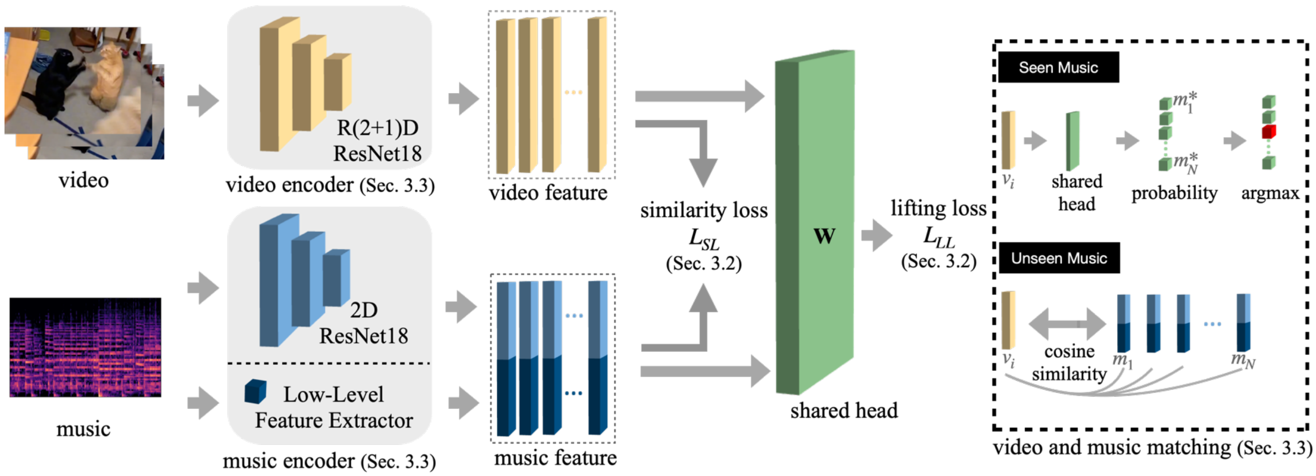

Figure 2. **Overview of the proposed VMCML framework** for video-music matching. Given a pair of video and music, the video encoder is applied to extract video features and the music encoder is adopted to collect music features(Sec. 3.3). There are two parts of the music encoder, and we concatenate their output as music features. In the training step, we calculate the cross-modality lifting loss for the video and music separately from video features and music features by utilizing the same shared head (Sec. 3.2). Hence, calculate the cross-modality similarity loss between video features and music features(Sec. 3.2). In the testing step (the right side of the figure), we treat the task as a classification problem on seen music set and calculate the cosine similarity between video and music on unseen music set to match video with appropriate music (Sec. 3.3).

limitation, we follow [21] to adopt "Cross-Modality Similarity Loss" aiming at addressing the video-to-music feature distances to improve our downstream video-music matching performance:

$$\begin{aligned} \cos(\mathbf{x}_i, \mathbf{x}_j) &= \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \cdot \|\mathbf{x}_j\|} \,, \\ L_{SL}(\mathbf{v}_i, \mathbf{m}_i, \mathbf{m}'_i) &= max[0, \cos(\mathbf{v}_i, \mathbf{m}'_i) - \tau] \\ &\quad + 1 - \cos(\mathbf{v}_i, \mathbf{m}_i) \,, \end{aligned} \tag{4}$$

where $(\mathbf{v}_i, \mathbf{m}_i)$ indicates a positive video-music pair queried according to the ground truth music label of $\mathbf{v}_i$, $(\mathbf{v}_i, \mathbf{m}'_i)$ indicates a negative one randomly sampled from the dataset, and τ is a selected margin value. With cross-modality similarity loss, we can apply direct constraints in-between the predicted video and music features, and consider modality-to-modality distances, which provides consistent video-music matching schemes under both training and testing stages.

3.3. VMCML Framework

Video Encoder. To extract video features, we adopt a R(2+1)D ResNet-18 [22], pretrained on Kinetics-400 [11] and followed by a fully-connected layer to infer the video features with embedding size l, as our video encoder.

Music Encoder. we firstly calculate the Mel spectrograms of the input music and adopt a ResNet-18 [7] pre-trained on ImageNet [3] to extract the high-level music features. In addition, inspired by [29], we use openSMILE [5] to extract the low-level music features, including MFCC, voice intensity, pitch, etc. The low-level and high-level features are then fused by concatenation and we infer the final music features by passing the fused features into an additional fully-connected layer with embedding size l.

Training and Testing. During each training iteration, we use a pair of video and music for calculating L_{LL} (Equation (3)). For L_{SL} (Equation (4)), we randomly sample a negative music for calculation. The final training objective is established as:

$$L(\mathbf{v}_i, \mathbf{m}_i, \mathbf{m}'_i) = L_{LL}(\mathbf{v}_i, \mathbf{m}_i, \mathbf{W}) + \beta L_{SL}(\mathbf{v}_i, \mathbf{m}_i), \mathbf{m}'_i), \tag{5}$$

where $\mathbf{m}'_i$ indicates a randomly-sampled negative music, and β is a hyper-parameter.

During the testing stage, the seen music and unseen music sets are evaluated with different schemes, as illustrated in the right part of Fig. 2. For seen music set, the video features are inferred from our video encoder, while the music features are directly pulled from the trained prototype since the trained prototype can represent the feature center of each training music, which can significantly reduce the inference time of unseen music features. For unseen music set, the music features are extracted from our music encoder.

Finally, we match the videos and music by calculating the cosine similarity between their features and select

Dataset	#Video	Training #Music	Training VpM	Validation #Music	Validation VpM	Seen Music #Music	Seen Music VpM	Unseen Music #Music	Unseen Music VpM
MSV	150000	265	400	265	20	265	80	125	140
Youtube-8M [1]	5678	4654	1	NA	NA	NA	NA	1024	1024

Table 1. **Details of datasets.** The music for training, validation and seen music set is the same. Because the video-music pair is one-to-one mapping in Youtube-8M [1] dataset, validation and seen music are not included. **VpM** indicates the number of matching videos for each music.

the top 20 music clips with the highest similarities as the matched music list.

4. Experiments

In this section, we describe the datasets including MSV and Youtube-8M [1] in Sec. 4.1. The implementation details and evaluation metric are presented in Sec. 4.2. Moreover, we provide quantitative and qualitative results for the comparison between our approach and state-of-the-art techniques in Sec. 4.3. The ablation study is in Sec. 4.4.

4.1. Dataset

We summarize the details about the datasets in Table 1 and further illustrate them as follows.

MSV. Matching short videos with background music requires a suitable dataset, but there are limitations of public datasets [1]. The TikTok Dataset [10] only provides videos without a music component. The Youtube-8M [1] dataset contains both video features and music features, but most of the music consists of the original sound of the videos and each piece of music is only matched to a single video. Additionally, most videos are longer than one minute. To overcome these limitations, we collect videos with background music from a well-known multimedia platform. We randomly download the music and filter out music that(1) is the original sound of the video or (2) was uploaded by the video uploaders. We randomly download the corresponding videos for each song and there are no gender or age restrictions for uploaders. Furthermore, we exclude videos longer than 20 seconds. The video resolution is down-sampled by one-fourth to reduce data size. Our Music for Short Video (**MSV**) dataset comprises approximately 150,000 video-music samples, each 8 seconds long, with one frame per second. We used the uploader's choice of background music as the ground truth for each sample. In summary, the dataset includes 390 music, split into a seen music set (265 clips) and an unseen music set (125 clips). The seen music set was randomly divided into training, validation, and testing sets, with a ratio of 20:1:4, respectively, for 500 corresponding videos. The unseen music set contains 140 videos for each music, with no music present in the training music set. We will release **MSV** dataset after the acceptance by the conference. Both video and music will be released as binary features, and the link to the original site is provided.

Youtube-8M [1]. Youtube-8M [1] is not ideal for our proposed use case since the majority of the music in video-music pairs is the natural sound from the video. We follow CEVAR [21] to conduct experiments on a random subset of 6000 clips. We use the pre-computed video-level features in the dataset: a single vector for audio information and a single vector for visual information in a video. All experiments use these fixed pre-computed features. Youtube-8M includes video genre classification labels that indicate the topic of the video clip, and CEVAR utilized the labels as additional training signals. We ignore the video genre classification labels to make the experiment setting the same as our MSV dataset.

4.2. Implementation Details

We implement our proposed framework using three NVIDIA RTX 3090 with PyTorch [18]. All of the network parameters, including video encoder, music encoder, and shared prototype, are jointly optimized by Equation (5). Adam [13] optimizer is adopted with learning rate 1e-5, and both the margin values μ and τ are set to 0.2. The weight decay is set to 0.002, and the batch size is set to 128 for all models. The embedding size l of video and music features is set to 256. Hyper-parameters are selected based on the evaluation metric on the validation set at recall@10. α and β in Equation (3) and (5) are 0.38 and 2, respectively.

Evaluation Metric. The evaluation metric Recall@K is denoted as:

$$\text{Recall@}K = \sum_{v \in \mathcal{V}^{te}} \dot{\#} Hits_v @K \tag{6}$$

[1]The dataset used in CMVAE is not publicly available.

	Seen Music				Unseen Music			
	Recall@1	Recall@5	Recall@10	Recall@20	Recall@1	Recall@5	Recall@10	Recall@20
VMCML$_{\mathbf{Cosface}}$	**0.2056**	**0.3562**	**0.4409**	**0.5450**	**0.0170**	**0.0678**	**0.1329**	**0.2526**
VMCML$_{\text{Arcface}}$	0.1685	0.2772	0.3370	0.4203	0.0145	0.0673	0.1298	0.2455
Random	0.0037	0.0188	0.0377	0.0754	0.0080	0.0400	0.0800	0.1600
CEVAR [21]	0.1883	0.3319	0.4188	0.5300	0.0156	0.0613	0.1153	0.2170
CMVAE [29]	0.0277	0.0817	0.1316	0.2040	0.0112	0.0474	0.0906	0.1740

Table 2. Comparison the performance (Recall@K) between the proposed VMCML with baseline methods on MSV. VMCML$_{\text{Cosface}}$ indicates that the cross-modality lifting loss of the framework is CosFace loss [24]; VMCML$_{\text{Arcface}}$ represents that the cross-modality lifting loss of the framework is ArcFace loss [4]. All architectures we implement use the same video encoder and music encoder.

	Seen Music				Unseen Music			
	Recall@1	Recall@5	Recall@10	Recall@20	Recall@1	Recall@5	Recall@10	Recall@20
ResNet-18	**0.2056**	**0.3562**	**0.4409**	**0.5450**	**0.0170**	**0.0678**	**0.1329**	**0.2526**
Vggish	0.1999	0.3453	0.4325	0.5358	0.0121	0.0574	0.1129	0.2155

Table 3. Comparison of Resnet18 and Vggish for music encoder backbone on VMCML architecture. Vggish is the backbone originally used by CMVAE as the music encoder.

where $\mathcal{V}^{te}$ denotes the set of testing video. The Recall@K indicates the percentage of queries for which the model returns the correct item in its top K result.

4.3. Experimental Results

Baselines. To demonstrate the effectiveness of the proposed framework, we compare our VMCML with the state-of-the-art video-music matching approaches. 1) CEVAR [21]: the approach adopting softmax loss for video-music matching. 2) CMVAE [29]: the approach adopting a variational auto-encoder for video-music matching.

Comparison on MSV. The quantitative results on our proposed MSV dataset is shown in Table 2. In general, our proposed method outperforms the other baselines in all metrics. Compared to CMVAE, we found that learning the distribution of videos and music with a variational auto-encoder might not be a satisfying solution for video-music matching task, and our method improved Recall@10 by 2.4% on seen music and 46.7% on unseen music . On the other hand, although CEVAR also learns a shared prototype with a softmax loss function, we improve Recall@10 by 5.3% on seen music and 15.3% on unseen music. We found that lifting video and music features to a hyper-sphere by our cross-modality lifting loss is necessary for improving the decision boundary between classes.

In addition to the comparison with SOTA approaches, we also compare the performance difference between CosFace and ArcFace loss functions. We empirically found that CosFace leads to a better training convergence, and thus

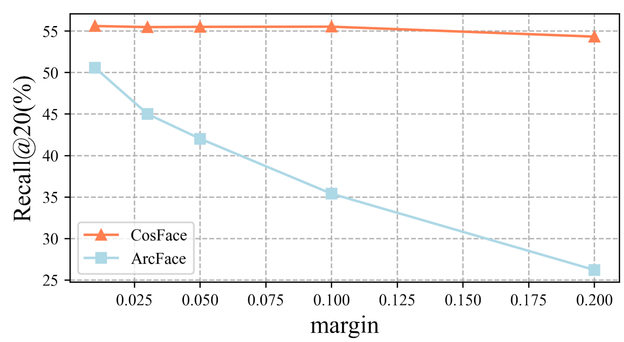

Figure 3. Recall@20(%) with different margin parameters μ on CosFace [24] and ArcFace [4]. We implement the experiments with VMCML on seen music set.

we adopt CosFace to lift our video and music features to a shared hyper-sphere. Moreover, we conduct experiments for selecting the best margin value μ in Equation (3) as shown in Fig. 3. We evaluate the recall@20 performance on the seen music set of MSVD by increasing μ from 0.01 to 0.2. The results indicate that CosFace achieves better performance and training stability than ArcFace does. As μ increases from 0.1 to 0.2, the variation of the ArcFace performance is approximately 0.1, while the variation of CosFace performance is smaller than 0.01. These findings suggest us to adopt CosFace in our cross-modality lifting loss.

Furthermore, as shown in Table 3, we compare the performance of two different backbones adopted in our music encoder. 1) Vggish: the backbone adopted by CMVAE. 2) ResNet-18, a more advanced residual network [7]. The performance of ResNet-18 outperforms Vggish at all metrics on both seen music and unseen music despite the fact that Vggish is pretrained on Audio Set [6]. However, the results of adopting Vggish in our VMCML framework still outperform other SOTA approaches on seen music set in Table 2.

Comparison on Youtube-8M [1]. We also evaluate our method on the Youtube-8M, a cross-modal video and music retrieval benchmark dataset, and compare it with SOTA approaches. Our evaluation is performed on the 1024 video-music pairs given by the repository of CEVAR. Since the features provided by Youtube-8M are one-dimensional, we use the original CEVAR backbone instead of R(2+1)D and 2D ResNet-18. The backbone consists of a set of fully connected layers that transform the original features into embeddings, with each hidden layer using ReLU as the activation function. Our results, presented in Table 4, demonstrate the effectiveness of VMCML for cross-modal video and music matching on both seen music and unseen music. Compared to CEVAR, our approach improves Recall@10 by a factor of 2 on unseen music. These results convincingly demonstrate better applicability of our proposed method than other methods.

	Unseen Music			
	Recall@1	Recall@5	Recall@10	Recall@20
VMCML	**0.0645**	**0.0879**	**0.1250**	**0.1807**
Random	0.0010	0.0049	0.0098	0.0195
CEVAR [21]	0.0105	0.0356	0.0621	0.1041
CMVAE [29]	0.0039	0.0147	0.0273	0.0469

Table 4. Comparison between the proposed VMCML with state-of-the-art method on Youtube-8M [1]. Because the video features and music features in Youtube-8M [1] are one-dimensional, the video encoder and music encoder used in this experiment are fully connected layers.

Qualitative Comparison. To further address the performance difference between our VMCML and CEVAR, we visualize the predicted top-5 matched music in unseen music for video in Fig. 4 on MSV dataset. The matched music is sorted based on similarity scores in descending order, with the ground truth one highlighted by red boxes. From the first example, we can observe that the VMCML method predicts the ground truth music with the highest similarity score, while CEVAR cannot even find the ground truth one in the top-5 matched list. Also, by visualizing the similarity of the prediction shown in Fig. 4 (c), we found that the similarity scores of CEVAR for each music only vary in a small range, while VMCML produces higher variances between different music.

4.4. Ablation Study

Low-Level Feature. We conduct an ablation study for the effectiveness of low-level music features extracted from openSMILE feature extraction toolkit, including CHROMA, loudness, and pitch. We compared the video-music matching performance with and without low-level features on both seen music and unseen music. The results are presented in Table. 5. The results suggest that low-level features are crucial for video-music matching tasks. Particularly on unseen music, the performance of Recall@5 is improved by 1.8 times due to the fact that the pitch of the music provides information about the tempo and speed, which are directly associated with the alignment between actions in videos with the drum beats in the music. Additionally, the changes in music volume with time are also an important cue for matching video and music, which further improves the matching performance. Furthermore, we conduct experiments involving low-level music features with and without high-level music features to assess their effectiveness. The results indicate that considering only low-level features negatively impacts performance. This occurs because our training method elevates video features and music features to the same hyper-sphere, wherein low-level features remain unchanged during training stage.

	Seen Music			
	Recall@1	Recall@5	Recall@10	Recall@20
w/ llf	**0.2056**	**0.3562**	**0.4409**	**0.5450**
w/o llf	0.1776	0.3109	0.3939	0.5042

	Unseen Music			
	Recall@1	Recall@5	Recall@10	Recall@20
w/ llf	**0.0170**	**0.0678**	**0.1329**	**0.2526**
w/o llf	0.0147	0.0381	0.1079	0.2087

Table 5. Compare VMCML performance on both seen and unseen music sets when training with and without music **l**ow-**l**evel **f**eatures (**llf**).

Similarity Loss. The effect of cross-modality similarity loss is shown in Table 6. The similarity loss aims at discriminating the features between a positive and a negative pairs of video and music inputs, and consider modality-to-modality distances for further improving matching performance. Hence, our results show that incorporating the similarity loss improves performance on both seen music and unseen music, indicating that video-music matching task can benefit from this technique.

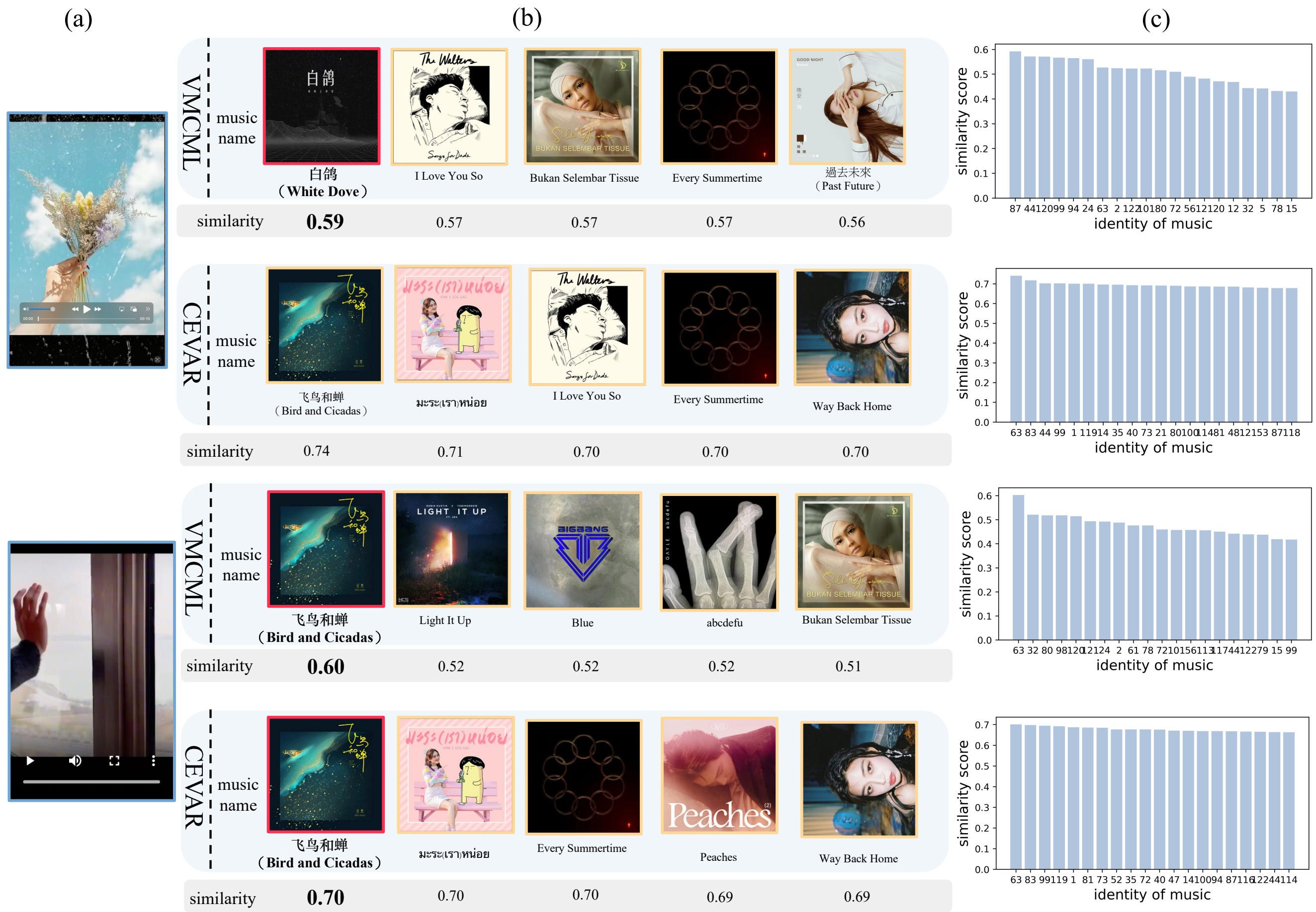

Figure 4. **Qualitative of VMCML for matching video and music** is visualized by two examples in unseen music set, and the comparison method CEVAR is shown. (a) are videos (in the blue box). (b) are the top five matching music (in the yellow box) ranked by their similarity score the video, with the ground truth is in the red box. (c) are the top-20 similarity score.

	Seen Music			
	Recall@1	Recall@5	Recall@10	Recall@20
w/ L_{SL}	**0.2056**	**0.3562**	**0.4409**	**0.5450**
w/o L_{SL}	0.1958	0.3206	0.3935	0.4840

	Unseen Music			
	Recall@1	Recall@5	Recall@10	Recall@20
w/ L_{SL}	**0.0170**	**0.0678**	**0.1329**	**0.2526**
w/o L_{SL}	0.0123	0.0596	0.1166	0.2186

Table 6. Comparison of the performance when training with and without similarity loss on VMCML.

5. Conclusion

In this work, we develop a cross-modal framework VMCML, which addresses the challenges in music recommendation for new users or new music give short-form videos. VMCML constructs a shared embedding space between video and music representations effectively by adopting CosFace loss based on margin-based cosine similarity loss. Also, we collect a large-scale dataset (**MSV**) which contains 390 individual music clips and the corresponding matched 150,000 videos. We demonstrate that our approach achieves state-of-the-art performance for matching video and music on **Youtube-8M** and our **MSV** datasets.

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. 2, 3, 5, 7

[2] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning

a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005. 3

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 3, 4

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699, 2019. 2, 6

[5] Florian Eyben, Martin Wöllmer, and Björn Schuller. Opensmile: the munich versatile and fast open-source audio feature extractor. In *ACM Conference on Multimedia (MM)*, pages 1459–1462, 2010. 4

[6] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017. 7

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4, 7

[8] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, pages 84–92. Springer, 2015. 3

[9] Sungeun Hong, Woobin Im, and Hyun S Yang. Cbvmr: content-based video-music retrieval using soft intra-modal structure constraint. In *ACM Conference on Multimedia (MM)*, pages 353–361, 2018. 2

[10] Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12753–12762, June 2021. 2, 5

[11] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4

[12] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4873–4882, 2016. 2

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[14] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *IEEE International Conference on Computer Vision (ICCV)*, pages 706–715, 2017. 2

[15] Bochen Li and Aparna Kumar. Query by video: Cross-modal music retrieval. In *ISMIR*, pages 604–611, 2019. 2

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 2

[17] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 212–220, 2017. 2

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[19] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, pages 2641–2649, 2015. 2

[20] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. *Advances in neural information processing systems*, 27, 2014. 2

[21] Didac Surís, Amanda Duarte, Amaia Salvador, Jordi Torres, and Xavier Giró-i Nieto. Cross-modal embeddings for video and audio retrieval. In *European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2, 3, 4, 5, 6, 7

[22] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[23] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*, pages 586–587. IEEE Computer Society, 1991. 2

[24] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 6

[25] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1386–1393, 2014. 3

[26] Jiwei Wei, Xing Xu, Yang Yang, Yanli Ji, Zheng Wang, and Heng Tao Shen. Universal weighting metric learning for cross-modal matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13005–13014, 2020. 2

[27] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *Computer Vision–ECCV 2016: 14th European*

Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14, pages 499–515. Springer, 2016. 3

[28] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016. 2

[29] Jing Yi, Yaochen Zhu, Jiayi Xie, and Zhenzhong Chen. Cross-modal variational auto-encoder for content-based micro-video background music recommendation. *IEEE Transactions on Multimedia (TMM)*, 2021. 2, 3, 4, 6, 7

[30] Donghuo Zeng, Yi Yu, and Keizo Oyama. Audio-visual embedding for cross-modal music video retrieval through supervised deep cca. In *2018 IEEE International Symposium on Multimedia (ISM)*, pages 143–150. IEEE, 2018. 2

[31] Liangli Zhen, Peng Hu, Xu Wang, and Dezhong Peng. Deep supervised cross-modal retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2