It’s Time for Artistic Correspondence in Music and Video

Didac Surís
Columbia University
didacsuris@cs.columbia.edu

Carl Vondrick
Columbia University

Bryan Russell
Adobe Research

Justin Salamon
Adobe Research

musicforvideo.cs.columbia.edu

Abstract

We present an approach for recommending a music track for a given video, and vice versa, based on both their temporal alignment and their correspondence at an artistic level. We propose a self-supervised approach that learns this correspondence directly from data, without any need of human annotations. In order to capture the high-level concepts that are required to solve the task, we propose modeling the long-term temporal context of both the video and the music signals, using Transformer networks for each modality. Experiments show that this approach strongly outperforms alternatives that do not exploit the temporal context. The combination of our contributions improve retrieval accuracy up to $10 \times$ over prior state of the art. This strong improvement allows us to introduce a wide range of analyses and applications. For instance, we can condition music retrieval based on visually defined attributes.

1. Introduction

Music is a crucial component of video creation, for example soundtracks in feature film, music for advertisements, background music in video blogs, or creative uses of music in social media. However, choosing the right music for a video is difficult—the video creator needs to determine what kind of music to use for different moments in the video and then search for this music. Each of these tasks presents difficulties: choosing the right music to set the mood of a video can be hard for non-professionals and, even when you know what type of music you want, it can be hard to search for it using conventional text-based methods. It is very hard to describe the “feel” of a song in words and metadata-based search engines are not well suited for this task.

An automated tool to suggest relevant music given video footage as input could be of great value to a range of creators, from beginners and amateurs in need of a simple solution, through to communicators and professionals in search of inspiration. The inverse problem, matching video...
footage to a given song, similarly presents notable challenges, and a solution has the potential to unlock new creative applications. As such, an automated tool to perform retrieval in both directions, from video to music and vice versa, is of great interest.

While other audio-visual tasks aim to establish physical correspondences for discrete events between the two modalities (e.g., the sound of a person clapping with the visual motion of the person performing the clapping action) [5–7], such correspondences are predominantly not the deciding factor for pairing music with video. The determining factors for the pairing task are instead often “artistic” and non-physical, and may comprise the overall visual style or aesthetics of the video, and the genre, mood or “feel” of the music. Additionally, a system may pair musical genres with visual attributes (e.g., depicted scene type or musical instrument played) or populations presenting a particular gender or race. Studying the interplay of these factors is important for understanding and exposing how a system makes its decisions and mitigating potential bias [26].

To address these tasks, we seek to train an audio-visual model to determine how well a paired video and music audio clip “go together”, or correspond, where we learn this correspondence directly from video data without requiring any manual labeling. Once trained, the model can be used to retrieve music that would pair well with a given input video, and to retrieve video clips that would pair well with a given input music track (see Fig. 1 for some examples). Moreover, we seek to understand how a trained model associates the aforementioned musical genres and visual attributes.

As it is difficult to manually collect annotated data at large scale describing the mood of video and musical audio, we leverage self-supervision, i.e., learning from the structure inherent to the data. Since we have access to large video collections where music and video have already been paired together by human creators, we leverage these data to learn what makes for a good pairing. The model is presented with both the large collection paired by human creators and randomly paired audio/video tracks, and is trained to distinguish between the two collections.

Previous approaches for this task typically rely on corresponding short video and music segments or aggregating features over multiple segments [51]. However, as the correspondence between video and music is often an artistic one, it often depends on long-range temporal context, which is hard to capture in a short segment or by aggregating multiple segment features. For example, a given scene in a movie conditions the “mood” of the music in the next scene, and the proximity of the climax of a song conditions how the video clip is edited [33, 42, 52].

Furthermore, these prior approaches optimize metric losses that do not weight hard examples during training [15], and leverage modality-specific visual base features trained on a fixed-vocabulary classification task [21] or audio base features that are not specific for music [17].

Finally, while these approaches evaluate retrieval accuracy, they do not study how a model associates musical genre and visual attributes.

To address these challenges, we make the following contributions. First, we show for the first time that temporal context is important for this artistic correspondence learning task. We do so by leveraging a Transformer architecture [63] to model long-range temporal context and employing other best practices for video-music retrieval (e.g., optimize a contrastive loss during training, build on strong base features for each modality), leading to a dramatic 10× improvement in retrieval accuracy. Second, we conduct a detailed analysis of our model, shedding light onto what visual attributes present in the video, such as scene type and musical instruments, are used by the model to establish artistic correspondence with different musical genres. This analysis includes “attributes” whose over-simplistic definition or representation such as gender and race can lead to potentially concerning biases. Third, we demonstrate the usefulness of the learned audiovisual representation through several applications, including novel ones such as combining a music query with visual attributes to retrieve music of the same genre where the visual attributes are musically represented in the audio signal. Finally, we study and discuss potential issues with our model related to bias. Since our task is concerned with learning artistic correspondence based on video-music pairings made by humans, rather than audio-visual correspondence grounded in physics, it presents new and important challenges and considerations concerning bias, cultural awareness and appropriation.

2. Related Work

Music from video. Several frameworks have been proposed to recommend music for a given video. However, most of them have limitations that we address in this paper. Heuristics-based approaches [38, 57] only consider the general mood of the music video and the user listening history. The mood categories are annotated independently for the two modalities, require manual annotations for every video and audio segment, and are restricted to a limited number of pre-defined discrete categories.

Cross-modal ranking losses for music and video [33, 42, 71] and learned audio features [51] have been used to obtain state-of-the-art (SoA) results. We build on top of them with three key contributions that lead to a tenfold improvement in retrieval accuracy: 1) propose a framework that models temporal context, 2) use a noise-contrastive loss [47], which has been shown to be better suited to self-supervised settings, and 3) use SoA feature extraction models.

Music synthesis, a task that is hard on its own, can also be conditioned on a given input video. Approaches like gen-
erating MIDI files by looking at fingers playing [29, 58], and directly generating sound (foley) using spectrograms [31] have been proposed, but they can only exploit low-level signals that do not capture any artistic aspect of the video. Using pre-trained music generation models, conditioning them on video [19], limits the audiovisual correspondence to a few pre-determined parameters (e.g., energy, direction, and slope), which cannot be learned in a self-supervised fashion.

Recent previous work [52] studied the relationship between some music traits (e.g., beats) and video editing operations (e.g., cuts) by interviewing professional editors and computing statistics on existing video data. They observe that some of the correspondences require contextual information; for example, some editors increase cutting video to the beat close to a climax moment in the video, or choose the video content to emphasize a musical climax. Such a finding suggests that there could be value in modeling temporal context for correspondence learning. Our paper shows that context is important quantitatively, while also showing how to best achieve this technically.

Long-form video has been modeled in the literature by first computing representations at different temporal locations, and then combining them, either through averaging or by learning a more complex combination of temporal features [66–68]. We propose to model the long-term temporal context in both the music and video modalities using Transformers [63], which use attention to model long sequences, and have become the SoA method for many NLP tasks in the past few years. Recently, they have been adapted to other domains such as images [14, 16, 22, 23, 43], videos [4, 8, 9, 11, 46, 65, 69], audio [32, 64], multi-modality inputs [28], and even modality-agnostic Transformers have been proposed [4, 35], all with significant success. Video Transformers take as inputs either pixels directly, or features from pre-trained networks (e.g., [46]). We build on top of the latter approach, and experiment with both convolutional [24, 41, 53] and Transformer-based [11] base features, both for the visual and music modalities.

Audiovisual self-supervised learning has been studied in a number of papers [3, 5–7, 48, 60, 70] that deal with physical events and sounds, such as the sound of dogs, cars, or musical instruments, or the location of the sound sources. However, these papers do not deal with the higher-level artistic music-video correspondence.

Music-conditioned video editing has mostly focused on synchronizing music with dancing videos. Approaches range from video resampling to fit the music [20] to directly generating pixels of people dancing, conditioned on static images and a music track (e.g., [27, 40, 54]). The focus on the dancing, while musically oriented, relies on low-level correspondence between music beats and human movement. This paper focuses on higher level correspondence, where emotions, story, and context are key factors, and are not considered in dancing videos.

3. Music Video Pretraining over time (MVPr)

In the following sections we describe our proposed method, Music Video Pretraining over time, or MVPr.

Inputs and outputs. During training, the inputs to the framework are a collection of video and music pairs $V$ (music videos), where the music and video have been artistically paired by human creators. Each raw music video is processed to obtain base representations for the visual track $x^v$ and a music track $x^m$. Additionally, each music video is divided into $L$ segments, of duration $t$. Correspondingly, these segments consist of a visual segment and a music segment. The division into segments allows us to 1) process the music video as a sequence, and thus exploit temporal context, and 2) make separate predictions for every segment, at a more fine-grained temporal resolution.

Our model takes $x^v$ and $x^m$ as inputs, and it outputs representations $y^v = f_v(x^v)$ and $y^m = f_m(x^m)$, respectively, where $f(\cdot; \theta)$ represent the functions whose parameters $\theta$ are optimized. See Fig. 2 for an overview of the framework.

Cross-modal self-supervision. The music and visual tracks in videos have a strong correspondence. The music that plays on top of the video is artistically related to the content of the video. We exploit this alignment as supervision: given a representation of a visual segment, our model is trained to predict the representation of the corresponding music segment, and vice versa.

The energy function we optimize computes a similarity between the representations of video and music segments, and encourages positive (corresponding) pairs to have a high similarity value, and negative (non-corresponding) pairs to have a low similarity value. In practice, this is implemented using the InfoNCE contrastive loss [47]:

$$
\mathcal{L}_{v \rightarrow m} = - \frac{1}{L} \sum_{i=1}^{L} \log \frac{\exp \left[ s(y^v_i, y^m_i)/\tau \right]}{\sum_{j=1}^{L} \sum_{l=1}^{L} \exp \left[ s(y^v_j, y^m_l)/\tau \right]},
$$

where $s(y^v, y^m)$ is the similarity function, which follows common practice we implement as the cosine similar-
Table 1. Segment-level retrieval results for MusicVid-YT8M. Each one of our contributions improves the accuracy of the model.

<table>
<thead>
<tr>
<th></th>
<th>Median Rank ↓</th>
<th>Recall ↑</th>
<th>V→M</th>
<th>M→V</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V→M</td>
<td>M→V</td>
<td>R@1</td>
<td>R@5</td>
<td>R@10</td>
</tr>
<tr>
<td>Baseline</td>
<td>349</td>
<td>277</td>
<td>0.55</td>
<td>2.16</td>
<td>3.83</td>
</tr>
<tr>
<td>+ CLIP and DeepSim features</td>
<td>176</td>
<td>107</td>
<td>3.06</td>
<td>6.31</td>
<td>9.69</td>
</tr>
<tr>
<td>+ Transformers (music time)</td>
<td>27</td>
<td>26</td>
<td>16.53</td>
<td>27.04</td>
<td>37.14</td>
</tr>
<tr>
<td>+ Transformers (visual time)</td>
<td>24</td>
<td>24</td>
<td>17.23</td>
<td>27.54</td>
<td>38.64</td>
</tr>
<tr>
<td>+ InfoNCE (MVPt, ours)</td>
<td>19</td>
<td>12</td>
<td>17.33</td>
<td>29.12</td>
<td>39.33</td>
</tr>
<tr>
<td>MVPt + X3D features</td>
<td>28</td>
<td>27</td>
<td>8.47</td>
<td>19.66</td>
<td>28.87</td>
</tr>
<tr>
<td>MVPt + TimeSformer features</td>
<td>40</td>
<td>36</td>
<td>6.81</td>
<td>16.87</td>
<td>25.83</td>
</tr>
<tr>
<td>MVPt + t = 4.5s</td>
<td>52</td>
<td>52</td>
<td>11.70</td>
<td>17.18</td>
<td>24.79</td>
</tr>
<tr>
<td>MVPt + t = 11s</td>
<td>7</td>
<td>6</td>
<td>28.97</td>
<td>47.71</td>
<td>65.82</td>
</tr>
<tr>
<td>Chance</td>
<td>1000</td>
<td>1000</td>
<td>0.05</td>
<td>0.25</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Table 2. Track-level retrieval results for MusicVid-YT8M. Each one of our contributions improves the accuracy of the model.

<table>
<thead>
<tr>
<th></th>
<th>Median Rank ↓</th>
<th>Recall ↑</th>
<th>V→M</th>
<th>M→V</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V→M</td>
<td>M→V</td>
<td>R@1</td>
<td>R@5</td>
<td>R@10</td>
</tr>
<tr>
<td>Baseline</td>
<td>234</td>
<td>98</td>
<td>0.76</td>
<td>3.42</td>
<td>5.90</td>
</tr>
<tr>
<td>+ DeepSim audio features</td>
<td>142</td>
<td>94</td>
<td>1.41</td>
<td>5.23</td>
<td>9.01</td>
</tr>
<tr>
<td>+ CLIP visual features</td>
<td>64</td>
<td>45</td>
<td>3.00</td>
<td>11.13</td>
<td>18.56</td>
</tr>
<tr>
<td>+ Transformers w/o position</td>
<td>24</td>
<td>21</td>
<td>5.99</td>
<td>23.20</td>
<td>38.33</td>
</tr>
<tr>
<td>+ Temporal embeddings</td>
<td>18</td>
<td>17</td>
<td>6.09</td>
<td>24.91</td>
<td>41.89</td>
</tr>
<tr>
<td>Chance</td>
<td>1000</td>
<td>1000</td>
<td>0.05</td>
<td>0.25</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Inference. At inference time, the model takes a video as input, splits it into segments, and computes contextualized features for all segments. For each visual segment, it recommends a music segment that matches both the content of the visual segment, as well as the contextual information around it. The music segments are selected from a pool containing all the available music segments in the (test) dataset, according to the similarity metric used during training. The music to video retrieval is done equivalently. See Appx. B for implementation details.

4. Retrieval Experiments

We show retrieval experiments in two different settings. In the first setting (“track level”), we retrieve an entire full-length music audio track given a full-length query video (and vice versa). This setting allows evaluating the quality of the retrievals at the level of an entire (untrimmed) video. In the second setting (“segment level”), we aim to evaluate a finer-grained alignment between the two modalities where we retrieve a short music segment given a short video segment. In both the segment- and track-level settings, the inputs to our model are the $L = 30$ segments comprising the complete music video.

Given a query visual track (or music track), we compute

\[ s(y_v, y_m) = \frac{y_v^T y_m}{||y_v|| \cdot ||y_m||}. \]

\( \tau \) is a hyperparameter that we set to $\tau = 0.3$, following [15]. $\mathcal{L}_{m \rightarrow v}$ is defined symmetrically, and the final loss is $\mathcal{L} = \mathcal{L}_{v \rightarrow m} + \mathcal{L}_{m \rightarrow v}$, which is used to train the model using stochastic gradient descent.

Contextual models $f_v$ and $f_m$. Music and video are not only signals with a strong temporal component, they are also synchronized: changes in one modality are temporally aligned with changes in the other modality. Therefore, temporal context heavily impacts audiovisual correspondence, and needs to be modeled accordingly. To do so, we use a Transformer network [63], whose attention mechanism computes how much each element of the sequence has to attend to every other element in the sequence. We append a [CLS] token to the input, to represent the full video.

Base features. In our experiments, we use deep precomputed base features that are obtained from the visual and music raw signals. This allows us to 1) build upon state-of-the-art models and leverage large-scale pretraining, and 2) lift the representation demands from the Transformer networks, allowing them to focus their representation power on modeling the temporal context and the cross-modal alignment. Specifically, we use CLIP [53] for visual features and disentangled music tagging embeddings (DeepSim) [41] for music features. We temporally average the base features extracted for every segment of duration $t$. 

4. Retrieval Experiments

We show retrieval experiments in two different settings. In the first setting (“track level”), we retrieve an entire full-length music audio track given a full-length query video (and vice versa). This setting allows evaluating the quality of the retrievals at the level of an entire (untrimmed) video. In the second setting (“segment level”), we aim to evaluate a finer-grained alignment between the two modalities where we retrieve a short music segment given a short video segment. In both the segment- and track-level settings, the inputs to our model are the $L = 30$ segments comprising the complete music video.

Given a query visual track (or music track), we compute
the feature distance to each music track (or video track) in a pool of $N$ target candidates not seen during model training, where we set $N = 2000$ in all of the experiments following the setup in Prétet et al. [51]. Only one of the candidates is the correct pair (ground truth). We then sort the candidates according to this distance value and use two different criteria to evaluate the success of the retrieval. Recall@K (the higher the better): we look at the K closest candidates and consider the retrieval successful if the ground truth pair is among those, and we report the percentage of successful retrievals in the test set. Median Rank (the lower the better): we return the position of the ground truth pair in the sorted list of candidates; we then report the median of the position values across the test set.

Our approach is general and adaptable to any video data that contains music. We evaluate our method on two different datasets. **YT8M-MusicVideo**: we leverage a set of 100k videos from the YouTube8M dataset [2] that are tagged as “music video”, with an average track duration of 4 minutes. We use segments of duration $t = 6.7s$ for this dataset. **MovieClips**: we collect all videos from the MovieClips YouTube channel [45]. From these videos, we select the parts that contain music consecutively for at least 20s. We did so by training a PANN model [37] on AudioSet [30] and used it to detect regions with music in the data. The final number of selected video tracks in the dataset is 20k and their average duration is 42 seconds, and we use $t = 3.3s$. See Appx. A for more information about the datasets’ statistics and creation process.

**Baseline.** We build our contributions on top of the prior SoA method of Prétet et al. [51]. They propose a similar framework, but train with a triplet loss instead of an InfoNCE loss, use an MLP model instead of a Transformer, and use ImageNet base features for video and OpenL3 [17] base features for music. We refer to this model as “Baseline”. The input to the baseline model is the average across time of all the base features for the track level setting, and the average of the base features over a single segment for the segment-level setting. Our re-implementation of the baseline yields a retrieval accuracy (9.86% track level Recall@10) on our test set, that is close to the results reported by Prétet et al. (12.10%) for the MVD dataset [56], which we did not have access to. MVD is a manually curated sub-sample of the YT8M-MusicVideo dataset, so similar (same parent dataset) but slightly better (curated for clean audio-visual corresponances) results are to be expected.

**Ablations.** In our results, we show how modifying each of the model components contributes to an increase in retrieval accuracy. Note that 1) the MLP baseline has access to the same set of base features for each modality, but the features are aggregated via an average-pool operation before being passed as input to the MLP; and 2) in our Transformer model we match the number of model parameters (5.5M) to the baseline MLP, so the model capacity is not an advantage of our method. When using Transformers, a temporal encoding is added to each segment input. This setup allows the Transformer to exploit temporal information on top of the contextual one. In the segment-level setting, using temporal encodings for both modalities can result in a learning shortcut, where the model learns to associate visual segments to music segments based on their position in the sequence. Therefore, in the segment-level experiments, we disable the temporal embeddings for one of either the visual or the music modality. We report results for both options, indicating which modality keeps the temporal encoding as “music time” or “visual time”. Note that the Transformer is still capable of using contextual information.

**Results.** We show segment-level results for YT8M-MusicVideo and MovieClips in Tab. 1 and 3, respectively, and track-level results for the YT8M-MusicVideo dataset in

---

The diagram in Figure 3 shows the accuracy of the transformer decreasing as we remove temporal context, indicating its importance.
Tab. 2. The results show how each one of our contributions improves the performance of the model. Specifically, the modeling of temporal context via a Transformer proves critical (rows 2 and 3 in Tab. 1 and row 4 in Tab. 2). Additionally, the results show that representing temporal information (on top of context) in the visual track (“visual time” in Tab. 1, row 4) is just slightly more beneficial than modeling time in the music track (“music time”, row 3).

Also, as shown in both Tab. 1 (rows 6 and 7) and 3 (row 4), using video-trained base features like X3D [25] or TimeSformer [10] base features, while being better than the baseline ones, result in worse performance than using CLIP base features (which are trained on images). We argue that the reason for this result is that CLIP has been trained to align images with natural language sentences with a larger and more generic corpus of data than the ImageNet, X3D and TimeSformer models, that have been trained on fixed-vocabulary classification tasks. Finally, using longer segments improves the segment-level retrieval, at the expense of having a less fine-grained temporal representation (row 9 in Tab. 1).

To study the importance of temporal context, we evaluate our model when the input sequences are shorter, given the same segment length $t$. As seen in Figure 3, the model’s accuracy decreases as the sequence length decreases, which demonstrates the importance of temporal context. In comparison, the MLP version is not even capable of exploiting long temporal contexts in the first place.

Finally, to show that our approach captures correspondence from an artistic level, we perform human experiments: given a query video (or music) segment, we ask humans to choose between a music (or video) segment retrieved by our model, and another one retrieved by the baseline. 71.4% of responses prefer our method over the baseline, validating our claim ($p$-value < 0.01) that our model is preferable from an artistic viewpoint. More details are provided in Appx. C.2.

5. Analysis and Applications

In this section, we probe what our music-video model has learned, showing that it learns to use a wide range of signals, from relatively low-level ones (like music tempo) to high-level ones (like music genre). Additionally, we qualitatively evaluate the retrieval soundness of our model and show that both retrieval directions return samples that match the query at a remarkable level, to the point that, ignoring lip-syncing, usually look and sound correct to the human eye (and ear). Finally, we show how we can condition the retrieval results to return samples that contain specific attributes, and visualize the attention in our model.

5.1. Quantitative Analysis

We consider eight audio, visual, or audiovisual attributes: color brightness and hue, tempo, background scene, musical instruments, age, race, and gender. We include the latter two, in particular, to allow us to study questions related to bias. We adopt the definitions for race and gender from work targeting fairness in machine learning [36]. We study how each one of them influences our model’s predictions. We perform all the analyses on the YT8M-MusicVideo dataset. See Appx. C for more details.

Color brightness and hue. For every frame in the test set, we modify its brightness by a factor of $r$. Then, we perform retrieval as explained in Section 4 and compute the average Recall@10 for different values of $r$. Surprisingly, the Recall@10 accuracy only decreases 1 point ($42.37\% \rightarrow 41.24\%$) for a brightness variation of up to 30% ($r \in \{0.7, 1.3\}$), so brightness does not play an important role in our model’s accuracy, suggesting it may be using higher-level visual clues, which we analyze next. Likewise, we analyze the importance of hue: while more significant than brightness, hue is not crucial to model performance.

Tempo. We time-stretch the query music signal by a...
factor of $r$ to modify its tempo (i.e., make it slower or faster). We then evaluate retrieval and compute the average Recall@10 for different values of $r$. When modifying the tempo at a rate of 30% ($r \in \{0.7, 1.3\}$), Recall@10 accuracy drops more than 5 points (42.37% → 36.96%). We show the curves of Recall@10 as a function of $r$ for both brightness, hue, and tempo in Appx. C.3.

The subsequent attributes are all evaluated in the same way. We select an image dataset that contains annotations about that attribute and is balanced across the attribute classes. We compute representations for all the images in the dataset using the visual branch of our model and use them as target candidates. Note that our base CLIP features operate at the image level; we pass in a single base CLIP feature for the image to our Transformer model. Then, we use the music segments in our test set as queries and return the top-1 retrieved image from the balanced dataset. Finally, we plot a matrix of (music genre)-(visual attribute), normalized for every music genre. The music genre annotations, collected using musiconn [50], are only used for analysis, not during training. Note that because the target retrieval (image) dataset is balanced, the preference for each one of the attribute values is fully determined by the model.

Gender. We use the FairFace dataset [36], which contains images of faces, categorized by genre, race, and age. We show results in Figure 4 and discuss potential bias in Section 6. It is worth noting that a bias exists, and it is coherent with what we would expect in the real world—like female images being associated with the genre “female vocalists”—and with the bias in the training data—like male images being associated with “hip-hop”, “hard rock”, or “heavy metal”. Less than half of the genres show a strong preference for one of the genders, meaning that the model often does not rely on this attribute to reason about genre.

Race. Using the FairFace dataset, we repeat the previous analysis and show results in Figure 4. As expected, given the observed bias in our training data (Appx. A.1), “hip-hop” is mostly associated with black people, while “country” is mostly associated with white people. We attribute some unexpected associations to the lack of representation of certain races in our training dataset.

Age. We found age is not as important for the model as other attributes, so we moved the analysis to Appx. C.3.

Visual Scene. We use the Places dataset [72], which contains images of 205 scene categories. We observe that the scene attribute is also correlated with music genre, albeit less than the previous attributes. For reference, we list the most and least commonly retrieved scenes\(^1\) and show the genre→scene table in Appx. C.3.

\(^1\)Most commonly retrieved: track outdoor, runway, stage indoor, music-studio, boxing ring, baseball field, stadium baseball, martial arts gym, shoe shop, ballroom; Least commonly retrieved: canyon, residential neighborhood, snowfield, arch, attic, desert vegetation, crevasse, fire escape, mausoleum, water tower

---

Figure 6. We test the YT8M-MusicVideo model on a set of casually captured videos outside of the dataset, and show how our model generalizes to scenes that do not naturally contain music.

Figure 7. Retrieval in the MovieClips dataset. We show video-to-music retrieval examples, and show how our model exploits emotion to make the correspondence.

Visual Objects (Musical Instruments). We use instrument images from the Open Images Dataset [39], and proceed as in the previous attribute studies. As exemplified by the attribute conditioning (description below) in Figure 8a and 8d, the model learns a strong and useful representation for some instruments (e.g., guitar, drums) in both the visual and musical modalities. The genre→instrument table, shown in Appx. C.3, shows a clear preference from most music genres to retrieve guitar images, especially for hard rock, heavy metal, and punk genres. This is to be expected because 1) guitars are the most common instrument in these genres and 2) the video clips associated with these genres consist mainly of people playing the song, as opposed to other genres where the content is more cinematic.

5.2. Qualitative Analysis and Applications

Retrieval examples. The best way to show the quality of our model is to put it to the test on real examples. We evaluate the model on visual and music segments obtained from YT8M-MusicVideo, shown in Figure 1, and MovieClips, in Figure 7. In Figure 6, we test the YT8M-MusicVideo-trained model on a set of casually captured videos outside of the YT8M-MusicVideo dataset and show how our model can generalize to scenes that do not naturally contain music.
Attribute conditioning. Knowing that our model captures a range of audiovisual attributes, we propose using their representations to condition the retrieval process. In order to find a representation $y_a$ of a specific attribute (e.g., guitar), we use an auxiliary dataset with labeled images and/or audios representing that attribute (e.g., images of guitars), compute their representations, and average them to obtain the representation of the attribute. We implement the conditioning by adding the representation $y$ of the query segment to the attribute one: $y_{\text{conditioned}} = y + y_a$. If we instead want to remove the attribute, we use a subtraction: $y_{\text{conditioned}} = y - y_a$. We can apply these operations multiple times, for attributes defined using either of the modalities. To deal with the potential out-of-distribution problem when conditioning on data from a different domain, we found that better results are obtained when instead of $y_a$ we use $y_a = y_a - \sum_{b \in D} b$, where $D$ is the dataset the conditioning images or music tracks were obtained from (e.g., a dataset with images of instruments).

Overall, this procedure gives rise to a variety of applications, ranging from video editing—where we want specific attributes to be present—to music or video search. We showcase this variety of approaches with some examples in Figure 8. For instance, given an input music track not containing an instrument, we can retrieve a similar music track that contains that instrument. This instrument can be defined through data visually, as in example a), or via music, as in d). We can also condition on language. Specifically, in example b) we list a set of music tracks with non-English vocals, and we use them to retrieve music that is similar in style to a query input, but in a non-English language. Interestingly, the model creates a good representation of English (and as a consequence, non-English), but it is less consistent when representing other languages, probably due to the high proportion of English music tracks in the dataset.

Attention. We visualize attention results from the visual Transformer $f_v$ in Figure 5. For every example, we plot the attention weights at every video segment (represented by an image frame), computed using attention rollout [1]. These visualizations show that the model pays more attention to visual segments that contain people explicitly playing instruments or singing, over more cinematic content.

6. Discussion and Limitations

The correspondence between music and video is an artistic one. Art, and as an extension culture, is intrinsically tied to concepts such as language, nationality, gender, and race. Computer vision unfortunately still does not have the tools to deal with them in a satisfactory way. The result is a framework that resorts to bias and all the known negative effects bias can have in real-world applications [34, 44, 59].

Unlike other recognition applications [18], however, in the context of artistic correspondence a framework that is invariant to these factors could lead to the erasure of cultural traits or to cultural appropriation. On the other hand, explicitly magnifying the ties between music and culture—as often done by the music industry [55]—can exacerbate certain biases or associations.

In this paper, we adopt a descriptive approach and present the correspondences the model is learning. We consider this paper an invitation for further study and discussion of the interplay between culture and bias in the context of artistic correspondence learning and the challenges it presents. These advances will require collaboration between computer science and sociology. The complex question of how to appropriately design such a system for real-world applications remains an open question.

Finally, an explicit definition and precise evaluation of such concepts is lacking in our field and in this paper, and is an interesting avenue for future work.
References


[23] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale Vision Transformers. In IEEE International Conference on Computer Vision (ICCV), 2021. 3


[38] Fang-Fei Kuo, Man-Kwan Shan, and Shih-Yin Lee. Background music recommendation for video based on multi-modal latent semantic analysis. In 2013 IEEE International Conference on Multimedia and Expo (ICME), pages 1–6, 2013. 2


Appendix

We divide the Appendix in three sections. First, in Appx. A we describe the datasets in more detail. Second, in Appx. B we provide more implementation details about the model. And finally, in Appx. C we describe the experiments setup in more detail, and provide all the results that were not provided in the main paper.

A. Datasets Details

A.1. YT8M-MusicVideos

In this section we give details about the order to study the distribution of music genres in our YT8M-MusicVideos dataset (in the music part), and of gender, race, and age (in its visual part).

We compute genre information using musicnn [50]. We predict a single category from the MSD dataset [12] for every music track, and show its distribution in Figure 9.

We study the gender, race, and age information using the FairFace dataset [36]. First, we collect a random subset of 2079 samples in our dataset, and we extract faces from the raw frames (we sample one frame per second in the selected videos) using a pre-trained face detection model. Second, we train a classification model on the FairFace dataset. Note that the FairFace dataset is uniform across races and genres, which means the trained model is not biased. However, we note that the test-time accuracy of our model is not perfect (72% accuracy for race, 93% for gender), which makes this analysis just orientative. Finally, we classify the previously extracted faces using this model, and represent each video with the most common category across all the faces extracted from it. If a video does not contain faces, we do not use it for the analysis. We show results in Figures 10 and 11. In those figures, the frequencies are normalized for every gender.

The distribution of races, genders and ages in the random subset of 2079 videos of our dataset is the following:

- **Gender:** {Male: 1368, Female: 711}
- **Race:** {Indian: 21, White: 1098, Middle Eastern: 283, Black: 414, East Asian: 192, Latino_Hispanic: 57, Southeast Asian: 14}

A.2. MovieClips

In this section, we describe the procedure we used to create the MovieClips dataset. First, we downloaded all the videos from the YouTube channel MovieClips [45], which contains 37k videos. From these videos, we selected the parts that contain music consecutively for at least 20s. We did so by training a PANN model [37] on AudioSet [30] and used it to detect regions with music in the data. Specifically, we consider a segment of audio a “music” one if any of the music-related classes in AudioSet is predicted with at least a probability of 0.3 (the probabilities for different classes are predicted independently). We consider music-related classes all the ones under the “Music” category in the AudioSet ontology. In order to filter out non-music audio, we additionally ignore the segments that (even though they may contain music) contain any other audio category with a probability of 0.2 or more. These predictions are done at a very small temporal resolution, resulting in small gaps of silence between music segments. In order to not treat these segments as separate, but as a single one, we finally process the binary music predictions using a morphologic closing of 3 seconds.

B. Implementation Details

In videos whose total duration is longer than $K \cdot t$, clips are sampled from a subset of the video, of duration $K \cdot t$. During training, this subset is sampled randomly, and at test time the subset is found deterministically by centering it in the middle of the video.

The Transformer architecture follows the same design as the Transformer Encoder in the original paper [63]. We use hidden dimensionality $d_h = 256$, two layers, and two heads. We adapt the dimensionality $d_{in}$ of the input features to the hidden size $d_h$ using a linear projection layer.

We train the model using backpropagation, and following the optimization parameters in [61]. Specifically, we train all the models using AdamW optimizer and a cosine
learning rate annealing strategy with an initial value of 1e-3. We set the total batch size to 2048. We implement our models using PyTorch [49], and train on 8 RTX 2080Ti GPUs for 2 days.

C. Other Experiments

C.1. Experiment Setup Details

In order to obtain audios of guitars (used in Figure 8 of the main paper), we use the MedleyDB dataset [13], which has separate music tracks for every instrument, for a series of 196 songs (including both version 1 and 2 of the dataset).

C.2. Human Experiment Details

Each human was shown 20 example video pairs for each task (video-to-music, and music-to-video), where the pair consisted of the result provided by our model and the result provided by the baseline. The two clips in the pair contained either the same video and different music (for the video-to-music task), or two different videos with the same music (for the music-to-video task). The task consisted of a binary choice between the two clips, according to the best fit between music and video in the clip. Every human was shown different examples from the test set.

We collected a total of 296 human evaluations for the video-to-music task, with 220 of them (74.3%) preferring our model’s result over the baseline. Similarly, we collected 372 human responses for the music-to-video task, with 257 (69.1%) preferring our model’s output. Using the binomial test for statistical significance, we obtain in both cases a p-value well below 0.01, validating the claim that our model is preferable to humans.

C.3. Quantitative Analysis Extended

In this section, we extend the analysis in Section 5.1 in the main paper (“Quantitative Analysis”).

First, we show the curves of accuracy versus change rate $r$ in color brightness, color hue, and music tempo in Figure 12. In the case of hue, which is not an intensity parameter, a change of $r$ is a change in the hue following the equation $h_{\text{new}} = (h_{\text{original}} + 360(r - 1)) \mod 360$, where $h$ represents hue, which is an angle with values in $[0, 360)$. This hue change formulation allows us to represent all three modifications in a common scale.

We also show the matrices relating age, visual scene, and visual objects (instruments) to music genre in Figures 13 to 15. For the visual scene analysis on Places, from the 201 we only show the classes for which the sum of all the genres provides at least a sum of 20%. To obtain the balanced dataset of musical instruments, we download the images in the Open Images Dataset [39] that contain bounding box annotations of musical instruments and crop the instruments in those images. Then, we randomly sample 50 images from every instrument class in order to have a balanced dataset, and proceed as in the other attribute studies.

We also implemented a same-genre baseline, where we assume the genre label for each video and music track is known (for the video, we use the genre of its associated GT
music). From the full set of target candidates, we take just the ones with the same genre as the query, and randomly select one as the prediction. The average R@10 values are 10.00% for segment-level (compared to our 42.37%), and 7.27% for track-level (compared to our 42.27%). This result shows that, while genre is an important attribute, our model captures a variety of other relevant audiovisual cues (on top of capturing genre information). Note that the model does not have access to the ground truth (GT) genre. Recognizing the genre is a hard task by itself, and only well defined for music (not for video).

C.4. Qualitative comparison to baselines

We perform some qualitative comparisons between the GT audios and videos, the suggested baseline, and our model results, on the examples shown in Figure 1 of the main paper. In the first video-to-music example, the ground truth genre is “rock”. Compared to the ground truth audio, the music retrieved by our method has a very similar tempo, an equal reliance on pronounced beats, and a very similar language (Spanish vs. Portuguese). In contrast, the same-genre baseline (a random retrieval of a rock track) returns a music track which, while having similar instrumental to the ground truth (electric guitar and drums), is otherwise very different in terms of mood (sadder, less active), tempo (much slower), and language (Slovenian), and is a poor overall fit for the query video. We generally found that the same-genre baseline, unlike our model, is incapable of capturing important attributes not directly related to genre.

C.5. t-SNE visualizations

We computed t-SNE visualizations [62], which show videos and music tracks are generally clustered by genre, although not exclusively. See Fig. 16.

C.6. Addressing Potential Learning Shortcut

Transformers take in a sequence of clips, instead of a single average of all clips. This makes them capable of obtaining information about the length of the sequence. Therefore, a potential learning shortcut Transformers may exploit is the use of sequence length as a signal to match music to a video (music associated to a specific video will have the same length than the video). In order to show that our Transformer model is not relying solely on this shortcut and it is mostly exploiting other cues, we run a test that controls for the sequence length. Namely, we constrain the candidate retrieval set to be the same length as the query by setting the similarity scores between different-length sequences to zero. This would be equivalent to detecting the shortcut and exploiting it with perfect accuracy.

We perform this test for the YT8M-MusicVideos dataset, at the segment level (equivalently to Table 1 in the main paper). The numbers are in Table 4. Overall, the numbers imply that 1) our model is not relying on this shortcut, and 2) even when controlling for the bias (and providing the MLP baseline with a tool that it cannot have by construction), the Transformer model is still significantly better than the MLP baseline, showing that modeling of temporal context in a non-trivial way is highly beneficial for the performance of the model.
Figure 14. Scene vs genre. This figure is equivalent to Figure 4 in the main paper, but showing visual scene targets, instead of gender or race. While hard to associate specific genres to specific scenes, we note that there is a correlation, and the model is picking up on some signal regarding visual scene.

Table 4. Track-level retrieval results for MusicVid-YT8M when exploiting the potential shortcut.

<table>
<thead>
<tr>
<th></th>
<th>Median Rank ↓</th>
<th>V→M</th>
<th>M→V</th>
<th>V→M</th>
<th>Recall ↑</th>
<th>M→V</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>R@1</td>
<td>R@5</td>
<td>R@10</td>
<td>R@1</td>
<td>R@5</td>
<td>R@10</td>
</tr>
<tr>
<td>Baseline + CLIP and DeepSim features</td>
<td>5</td>
<td>5</td>
<td>20.31</td>
<td>50.96</td>
<td>67.43</td>
<td>23.49</td>
<td>55.52</td>
</tr>
<tr>
<td>MVPt (ours)</td>
<td>1</td>
<td>1</td>
<td>48.07</td>
<td>85.44</td>
<td>90.13</td>
<td>48.08</td>
<td>85.74</td>
</tr>
<tr>
<td>Chance</td>
<td>48</td>
<td>48</td>
<td>3.01</td>
<td>9.86</td>
<td>17.40</td>
<td>3.01</td>
<td>9.86</td>
</tr>
</tbody>
</table>
Figure 15. Instrument vs genre. This figure is equivalent to Figure 4 in the main paper, but showing instrument targets, instead of gender or race. The most retrieved instrument is the guitar, suggesting that the model has a good representation of guitars, and therefore we can use guitars for our conditioning experiments.

Figure 16. We project the music features into a 2-dimensional space using t-SNE [62], and color the data points by genre label. Music tracks are generally clustered by genre.