This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

MuseChat: A Conversational Music Recommendation System for Videos

Zhikang Dong [*]
Stony Brook University
zhikang.dong.1@stonybrook.edu

Xiulong Liu [*]
University of Washington
xl1995@uw.edu

Bin Chen
Bytedance.com
chen.bin@bytedance.com

Paweł Polak
Stony Brook University
pawel.polak@stonybrook.edu

Peng Zhang
Bytedance.com
zhang.peng@bytedance.com

Abstract

Music recommendation for videos attracts growing interest in multi-modal research. However, existing systems focus primarily on content compatibility, often ignoring the users' preferences. Their inability to interact with users for further refinements or to provide explanations leads to a less satisfying experience. We address these issues with MuseChat, a first-of-its-kind dialogue-based recommendation system that personalizes music suggestions for videos. Our system consists of two key functionalities with associated modules: recommendation and reasoning. The recommendation module takes a video along with optional information including previous suggested music and user's preference as inputs and retrieves an appropriate music matching the context. The reasoning module, equipped with the power of Large Language Model (Vicuna-7B) and extended to multi-modal inputs, is able to provide reasonable explanation for the recommended music. To evaluate the effectiveness of MuseChat, we build a large-scale dataset, conversational music recommendation for videos, that simulates a two-turn interaction between a user and a recommender based on accurate music track information. Experiment results show that MuseChat achieves significant improvements over existing video-based music retrieval methods as well as offers strong interpretability and interactability. The dataset of this work is available at https://dongzhikang.github.io/musechat.

1. Introduction

Music is an essential component in videos, enhancing both the viewer's experience and their understanding of the content. While existing music recommendation systems are proficient at selecting tracks that align with a video's theme—such as scary music for a horror film or upbeat tunes for a dance clip—these systems often neglect individual user preferences. For example, an '80s enthusiast may favor synth-pop over modern pop music for a nostalgia-themed video, even though both are categorized as "pop."

The challenge of personalized recommendation remains relevant, as many systems leverage user profiles and activity data to generate recommendations. However, we identify two key limitations: (1) the inability to consistently meet user preferences, and (2) the cold-start problem for new users without prior data. Current music recommendation systems aim to provide lists of songs based on user history, but these may not always align with user needs for specific videos. This not only affects user experience but also underscores the complexity of predicting preferences, which may deviate due to factors like user's recent trends. We propose a feedback mechanism to mitigate these limitations. This would allow the system to adjust its recommendations according to user feedback, aligning more closely with changing preferences. For new users without historical data, a cold-start scenario arises, leading to content-driven recommendation once again.

In this study, we introduce MuseChat, a comprehensive conversational music recommendation system for videos. As shown in Figure 1, MuseChat consists of two main modules: the music recommendation module and the sentence generator module. Users begin with uploading a video and receive a music recommendation tailored to the video's content. MuseChat enables user-system interactions through dialogues in natural language. At each dialogue turn, users are empowered by the music recommendation module in MuseChat. They can refine recommendations by specifying their preferences in natural language, such as mood, genre, instruments, theme, and artist details. This process continues until they identify their desired music track. Another distinguishing feature is the system's interpretability versus the "black box" nature of conventional music recommenda-

[*]Equally contributed. Work partially done during the internship at TikTok.

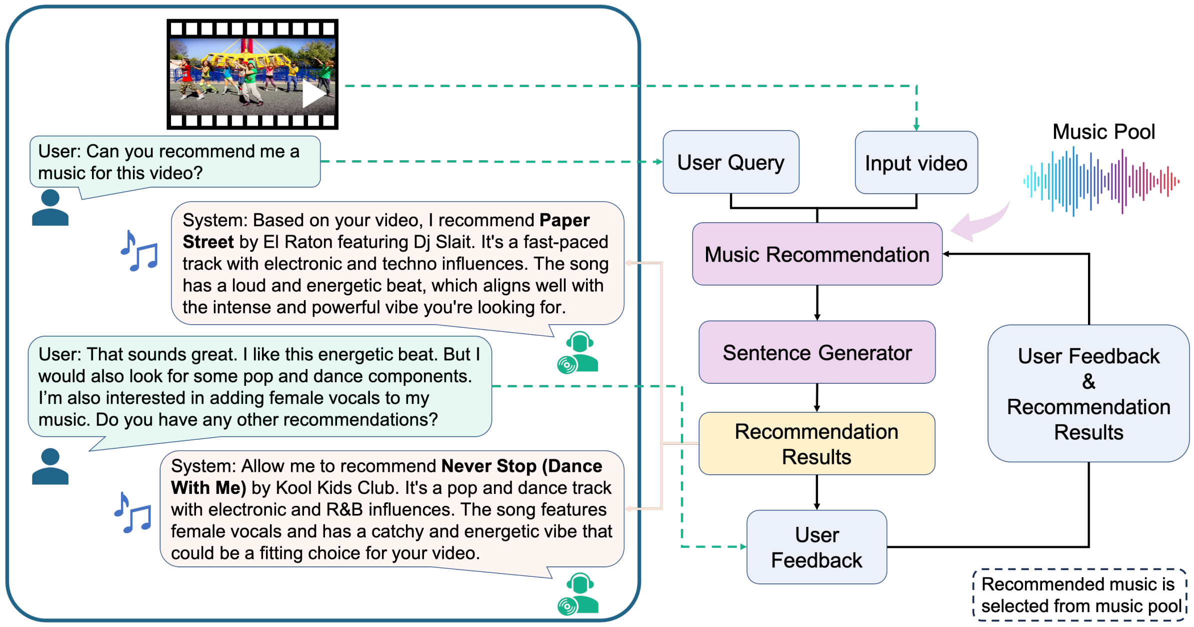

User: Can you recommend me a music for this video?
System: Based on your video, I recommend **Paper Street** by El Raton featuring Dj Slait. It's a fast-paced track with electronic and techno influences. The song has a loud and energetic beat, which aligns well with the intense and powerful vibe you're looking for.
User: That sounds great. I like this energetic beat. But I would also look for some pop and dance components. I'm also interested in adding female vocals to my music. Do you have any other recommendations?
System: Allow me to recommend **Never Stop (Dance With Me)** by Kool Kids Club. It's a pop and dance track with electronic and R&B influences. The song features female vocals and has a catchy and energetic vibe that could be a fitting choice for your video.
User Query
Input video
Music Pool
Music Recommendation
Sentence Generator
Recommendation Results
User Feedback
User Feedback & Recommendation Results
Recommended music is selected from music pool

Figure 1. MuseChat features two modules: the Music Recommendation Module, which processes either video input alone or in combination with user prompts and past music suggestions, and the Sentence Generator Module, which uses these inputs to create natural language music recommendations.

tion models. Our sentence generator module can not only justify the recommended music with reasons but also craft personal narratives for users based on the selected music.

Building upon this framework, our contributions are as follows: (1) We introduce a large-scale dataset tailored for a novel task, dialogue-driven music recommendations and reasoning within the context of videos. The data contains 98,206 quartets: a video, original music, candidate music and a two-turn conversation. This setup mimics the user's interaction with recommendation systems. It starts with uploading a video, receiving an initial music recommendation, and then accommodating a user's textual prompt to finalize the music selection; (2) We present a tri-modal architecture designed for music-video matching that incorporates with textual input. This model not only processes the previously recommended music and video content but also refines recommendations based on user-provided textual prompts; (3) We augment our model with the capability to offer clear, logical explanations for its music recommendations, achieved through the deep understanding of musical features by our LLM-based sentence generation module.

2. Related Work

Automatic music tagging. Music tags efficiently summarize songs by providing descriptive keywords that cover various elements such as emotion, genre, and theme. Numerous studies dive into the domain of automatic music tagging, as evidenced by works such as [4, 5, 28, 29, 38, 48]. Specifically, [49] employs a model that uses shallow convolutional layers to extract acoustic features, which are then processed by stacked self-attention layers in a semi-supervised setting. Similarly, [57] introduces S3T, a self-supervised pre-training method based on the Swin Transformer [32] architecture, further optimized by a music-specific data augmentation process.

Music description in free-form natural language. Describing music in free-form natural language has also gained research attention [22, 23, 34, 48]. For instance, [11] proposes a universal retrieval system to handle both tag- and sentence-level inputs. This system demonstrates adaptability across nine different music classification tasks. Moreover, [35] introduces "Song Describer," an open-source tool designed to gather text descriptions of music tracks from users. This initiative has led to the creation of a public audio-caption dataset in the music domain.

Music recommendation for video. The task of music recommendation based on video attributes receives attention in recent studies [39, 42, 52, 53]. While some works focus on creating joint embeddings of music and free-form natural language [23, 34], other studies examine the relationship between video, everyday audio sounds (excluding music), and language [18, 50]. [36] develops a method for guiding music recommendations with a single text description of music attributes. However, their approach does not incorporate user feedback, nor does it adapt its recommendations based on such feedback or prior recommendation results. Our work with MuseChat aims to address these limitations.

Conversational recommendation system. Conversational Recommender Systems (CRS) gain research attention for their ability to support task-oriented, multi-turn dialogues with users [8, 14, 16, 24, 51]. These systems capture the user's detailed and current preferences, provide explanations for suggested items, and process user feedback on recommendations. The emergence of LLMs substantially enhances the capabilities of CRS, particularly in understanding and generating natural language. [47] proposes an interactive evaluation approach that balances the focus between matching ground truth and maintaining interactivity. [13] utilizes LaMDA [43] to create a YouTube video recommendation system, relying on textual metadata instead of visual information.

Multi-modalities and Large language models. The rapid evolution of LLMs becomes a game-changer in the landscape of artificial intelligence, becoming a focal point in contemporary research. Originating from transformer architectures [46], these models train on extensive corpora, containing billions of words [10, 40]. Noteworthy models like OpenAI's GPT-3 [3], Meta's LLaMA [45], and Google's LaMDA [43] set benchmarks for textual generation that closely resembles human articulation. Moreover, with the increasing importance of computer vision and other modality tasks [7, 12, 25–27, 59], recent advances in LLMs extend beyond text-only input to incorporate multi-modal capabilities. These models are proficient at synthesizing and interpreting information across different data types. For instance, [60] introduces MiniGPT-4, which incorporates a visual encoder into a large language model. This leads to the model's ability to generate narratives inspired by images. Other notable works include [54], which can interpret video content to generate informed textual responses, and [30], which demonstrates how to encode answer candidates into GPT-3 prompts, enabling external knowledge integration. [31] proposes a novel method to answer audio-visual related questions in their balanced audio-visual-text dataset. Recent developments such as fine-tuning adapters [9, 15, 21, 55] make it easier for smaller research groups to adapt large models for specific uses, overcoming the high computational costs.

3. Dataset

We create the conversation part of the dataset by simulating a two-turn dialogue. In the first turn, the user provides only the video, and then a pretrained video-to-music retrieval system suggests a candidate music. In the second turn, based on this candidate music, the user provides more specific preferences in natural language as a prompt to guide the recommendation system. With the video, the candidate music, and the prompts, the system suggests the target music. While we only focus on generating two-turn dialogues here, we claim that our approach could easily extend to more dialogue turns. This is because our approach can treat subsequent dialogue turns as a "second turn", continuously adapting recommendations based on the previous result and current user instructions. Our approach involves the following steps, as Figure 2 shows: (1) collecting a large number of video and music pairs from an existing data source; (2) using a pretrained video-to-music retrieval model to select a candidate music for each video; (3) gathering music tags and other metadata for both the original and candidate music, which are then used to construct prompts; (4) feeding these prompts into GPT-3.5 for dialogue generation. We illustrate each step separately below.

Video-music pairs collection. We use the YouTube8M dataset [1] to create our conversational music recommendation dataset. YouTube8M is a large-scale labeled video dataset, encompassing millions of YouTube video IDs and relevant features for audio and visual content. It comes with thousands of labels covering a wide range of categories, such as music, sports, documentaries, and more. We select videos labeled as "music video" and remove unavailable videos, resulting in a collection of 98,206 music videos. From each music video, we extract a 120-second clip from the center. We choose this 120-second segment for three reasons: (1) It often captures the core information of the video, serving as a representative sample for music recommendation; (2) It minimizes noise from musical elements like intros and outros that could negatively impact the recommendation process; (3) It reduces computational costs.

Preparing candidate music. We implement a video-to-music retrieval model, named as Music-Video Pretrained (MVP) model, to retrieve an appropriate candidate music from the music pool to pair with the video. The MVP is a two-tower model, sharing a similar architecture with the models described in [23, 42]. It takes raw video and music clips as inputs, utilizing the pretrained CLIP Image encoder [41] for video feature extraction and the pretrained Audio Spectrogram Transformer (AST) [17] for music feature extraction.[1] This model is trained on our proprietary dataset consisting of 3 million high quality music-video pairs. More details for MVP are included in supplementary materials. We randomly divide the 98,206 music tracks from the music video clips into non-overlapping pools, with each containing 2,000 tracks. We employ the MVP model to select the candidate music from the same pool in which target track resides but excludes the target track. We empirically find that setting the pool size to 2000 could ensure the quality and diversity of candidate music as well as stay different from the target track. The MVP model computes

[1]The pretrained weights used are clip-vit-large-patch14 for the CLIP Image encoder and MIT/ast-finetuned-audioset-10-10-0.4593 for the AST.

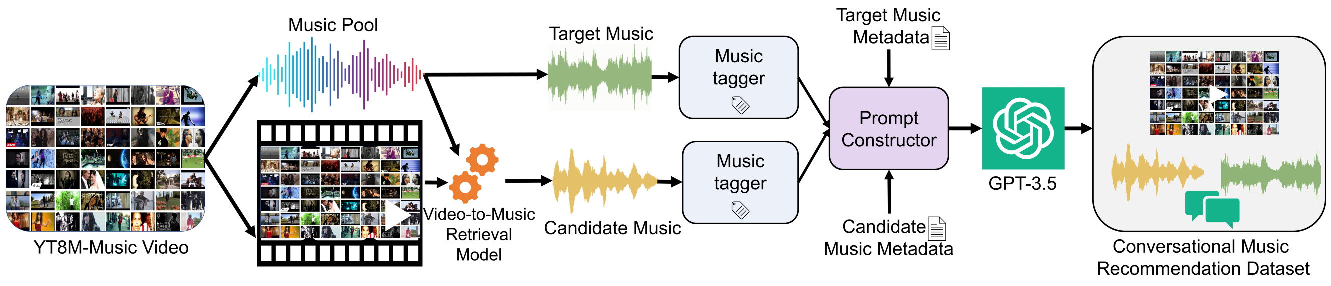


Figure 2. The generation pipeline for Conversational Music Recommendation Dataset.

the similarity between the input video and each track in the pool, ranking the tracks by descending similarity to determine the candidate music.

Constructing prompts. Our objective in this step is to construct prompts that effectively incorporate both the target and the candidate music tracks. Since each video from YouTube8M "music video" category has a paired music, we use it as the target music track in our simulated dialogue, towards which the user prompt would "steer" the recommendation from the candidate music. We employ the music tagging method from [38] to assign top 5 tags to each target music track. We leverage two distinct tagging systems for this purpose: one from the MagnaTagATune (MTT) dataset [19] and another from the Million Song dataset (MSD) [2]. Each system has a vocabulary of 50 tags, and the use of both systems increases the robustness of the tagging. The details and statistics of music tags in our dataset are provided in the supplementary material. Alongside music tags, we also collect metadata for music videos, which includes video title, video description, track name, artists, albums, and etc. These metadata are downloaded from the YouTube website for corresponding videos. Although each music video has a title and description, supplementary details such as official artist names, album specifics, and release dates are available for only around 30,000 tracks. We *manually* create various prompt templates using music tags and metadata from both original and candidate music tracks to diversify the synthetic conversations.

Dialogue turn generation. In the final stage, we generate dialogues that bridge the gap between the target music and candidate music. Utilizing GPT-3.5, we process the prompts constructed in the previous step. We observe that GPT-3.5 can effectively leverage its extensive knowledge base to enrich the conversations, even when the music metadata is unavailable. This integration of our carefully crafted prompts and GPT-3.5's generative capabilities ensures that our training data is both high-quality and varied, successfully imitating human interactions in the music recommendation context. The example prompt is available in the supplementary materials.

4. Approach

We propose a novel approach to address the conversational music recommendation task on this dataset, establishing a new baseline performance. As shown in Figure 1, there are two main modules involved in the system: music recommendation and sentence generator. We illustrate them below.

4.1. Music Recommendation

The goal of the music recommendation module is to select the most relevant music from the music pool using video, candidate music, and user text prompt. Each training sample is defined as a quartet (v, m_c, m_t, t_3), where v is the video, m_c denotes the candidate music track, m_t is the target music track and t_3 is the text of user prompt showing preferences. As illustrated in Figure 3, we focus on enhancing the model's ability to transit the recommendation from the existing candidate music m_c to target music m_t. Intuitively, when user prompt is given as an additional context to the candidate music and original video, the information combined from these modality contexts should arrive at a representation staying as close to the target music as possible. This motivates our formulation of the recommendation module as a metric learning problem: To learn a common embedding space between the tri-modal combination (candidate music + video + user prompt) and music. Towards this end, we propose a novel MVT-Fusion Module, as shown in Figure 3 (right), that explores music-text-video fusion in a fine-grained manner to obtain the tri-modal embedding. The embedding then serves as a goal embedding with which the target music is aligned. We utilize contrastive learning for the embedding alignment. We detail MVT-Fusion Module and contrastive formulation respectively.

MVT-Fusion Module takes the candidate music m_c (represented as mel-spectrogram), user prompt t_3, and video frames v as inputs. Each input is encoded via pretrained encoder in their corresponding modality. The candidate music is encoded using an AST encoder [17] g^{m_c}.

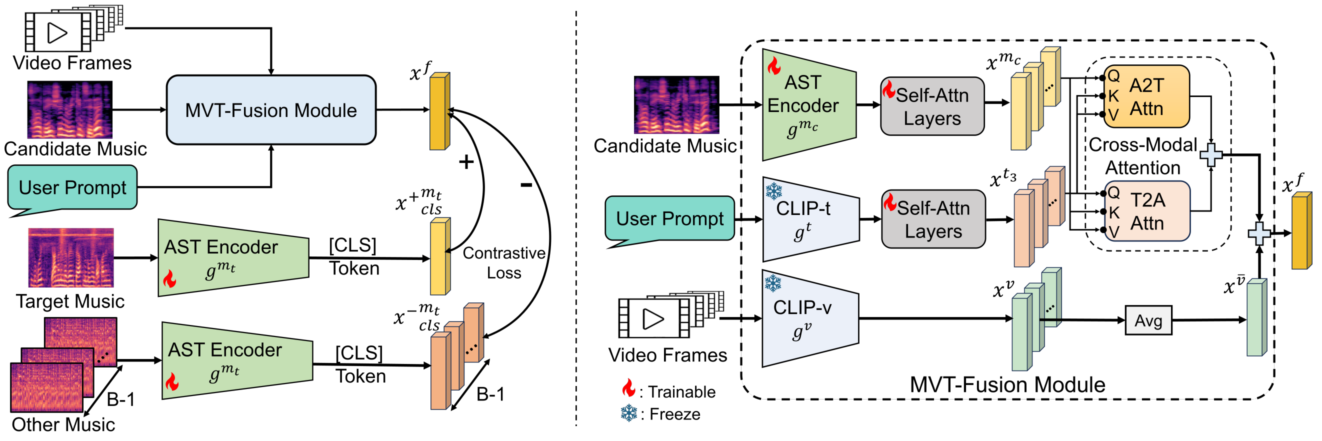


Figure 3. The music recommendation module combines video, the candidate music (1st round result), and user prompt (2nd round input) to retrieve a music in a common embedding space, trained using multi-modal contrastive loss (left). The candidate music corrected by user prompt along with the original video should result in a representation "closer" to the target music than other music. MVT-Fusion Module (right) is designed to combine the 3 modalities into an embedding space: i). The candidate music is encoded using the Audio Spectrogram Transformer (AST) [17]. ii). User prompt is fed into CLIP [41] text encoder (freeze) to get unpooled features. iii). Average pooling is performed on CLIP (freeze) vector of each video frame to obtain the video representation. iv). To foster fusion between candidate music and user prompt, self-attention layers and cross-modal attention (A2T and T2A, 'T' - text, 'A' - audio) are added to obtain the fused music-text vector.

The user prompt is fed into CLIP [41] text encoder g^t to obtain contextualized language features (unpooled). Each video frame is transformed into a vector using CLIP image encoder g^v, and is then averaged along the temporal axis to obtain a single vector $\mathbf{x}^{\bar{v}}$, representing the semantics of the video.

To better capture the correlation between the candidate music and the user text prompt, we develop a fusion method for merging the features from music and text modalities. We adopt the "late fusion" strategy, applying several self-attention layers to both output features from the CLIP text encoder and the AST encoder before fusion. The output features from self-attention layers of two branches are $\mathbf{x}^{t_3} \in \mathcal{R}^{n_t \times d}$ and $\mathbf{x}^{m_c} \in \mathcal{R}^{n_m \times d}$, and expanded as:

$$\begin{aligned} \mathbf{x}^{t_3} &= \left[x^{t_3}_{\text{cls}}, x^{t_3}_1, \dots, x^{t_3}_{(n_t-1)}\right], \\ \mathbf{x}^{m_c} &= \left[x^{m_c}_{\text{cls}}, x^{m_c}_1, \dots, x^{m_c}_{(n_m-1)}\right], \end{aligned} \tag{1}$$

where variable with subscript cls serves as their summary of the respective sequence, along with the other elements capturing detailed features.

Then, we fuse the transformed features by implementing a cross-modal attention layer, which is defined as: $\text{Att}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$, where d_k is the dimensionality of key vectors. Q and K, V are from two different modalities.

We add the video vector $\mathbf{x}^{\bar{v}}$ to obtain the tri-modal fusion vector representation $\mathbf{x}^f$:

$$\mathbf{x}^f = \mathbf{x}^{\bar{v}} + \text{Att}(x^{t_3}_{\text{cls}}, \mathbf{x}^{m_c}, \mathbf{x}^{m_c}) + \text{Att}(x^{m_c}_{\text{cls}}, \mathbf{x}^{t_3}, \mathbf{x}^{t_3}) \tag{2}$$

Contrastive Formulation Once $\mathbf{x}_f$ is obtained as the tri-modal fusion feature, it is necessary to align the representation of the target music with it. To achieve this, we first transform music from the pool into a vector space using a separate AST encoder g^{m_t}. Then we propose a contrastive learning approach to learn a common vector space between tri-modal fusion vector and music vector. Specifically, we use the hidden state of the last layer corresponding to the cls token as the music vector. We denote the target music vector as $\mathbf{x}_+^{m_t^{cls}}$, and any other music vector as $\mathbf{x}_-^{m_t^{cls}}$. Then, we formulate a contrastive loss aiming at keeping the vector distance between $\mathbf{x}_f$ and $\mathbf{x}_+^{m_t^{cls}}$ closer while pushing away $\mathbf{x}_-^{m_t^{cls}}$ from any other music, as illustrated in Figure 3 (left). Specifically, we use the Contrastive Multiview Coding Loss [44], a cross-modal variant of InfoNCE [37]. For each batch B, we have:

$$\mathcal{L}_{\mathcal{R}} = -\sum_{i=1}^{B} \left[\log \frac{h\left(\mathbf{x}^f_{(i)}, \mathbf{x}^{m_t^{cls}}_{(i)}\right)}{\sum_{j \neq i} h\left(\mathbf{x}^f_{(i)}, \mathbf{x}^{m_t^{cls}}_{(j)}\right) + h\left(\mathbf{x}^f_{(i)}, \mathbf{x}^{m_t^{cls}}_{(i)}\right)}\right], \tag{3}$$

where $\mathbf{x}^f_{(i)}$ and $\mathbf{x}^{m_t^{cls}}_{(i)}$ are i-th fusion vectors and target music vectors in the batch respectively. $h(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{\mathbf{x}^\top \mathbf{y}}{\tau}\right)$ is a discriminating function, with τ being a trainable temperature hyperparameter.

4.2. Sentence Generator

To equip MuseChat with reasoning capability, we propose a sentence generator to provide justification for the recommended music. Towards this end, we design a multi-modal LLM, as illustrated in Figure 4, using Vicuna-7B [58] (derived from fine-tuning Llama2-7B [45] model) as the backbone. Each training instance comprises a music representation, denoted as $\mathbf{x}^{\bar{m}_t}$, which is the average of the music embedding $\mathbf{x}^{m_t} = [x_{\text{cls}}^{m_t}, x_1^{m_t}, \ldots, x_{(n_m-1)}^{m_t}]$, derived from the previously fine-tuned AST Encoder g^{m_t}. The instance also includes the corresponding recommendation reasoning statement, t_4, from the simulated conversations. Notably, in our settings, each piece of music, whether as a candidate or target, has a corresponding reasoning statement in a conversation. Therefore, we do not specifically use the reasoning statement of candidate music in the first turn, t_2, because when this candidate music serves as the target in the other conversation, its reasoning statement is already utilized. We apply a linear layer f_l to project the averaged music features $\mathbf{x}^{\bar{m}_t}$ onto the text embedding space. To increase training efficiency, we leverage LoRA [21] to fine-tune the attention and output layers of Vicuna. We use next token prediction task aiming at minimize the negative log-likelihood of response tokens conditioned on the recommended music:

$$\mathcal{L}_{\mathcal{G}}(\mathbf{y};\theta) = -\sum_{i=1}^{n} log \left[p_\theta \left(y_i \mid \left[f_l \left(\mathbf{x}^{\bar{m}_t} \right), y_{<i} \right]; \theta \right) \right], \tag{4}$$

where y_i is the i-th token in the response $\mathbf{y}$, and θ is the trainable parameters.

Although we train the sentence generator using only music embeddings, during inference, we also input the title of suggested music from the music recommendation module. This is necessary because the model cannot generate accurate names of unseen music tracks.

5. Experimental Results

5.1. Implementation Details

We allocate 88,000 of music video clips to our training set. For each 120-second music video clip, we divide it into twelve 10-second segments and capture 5 frames per second from each segment. In our training process for the music recommendation module, each training sample includes a 10-second video clip, a corresponding 10-second target music clip, a 10-second candidate music clip, and a user prompt. To extract video and text features, we use OpenAI's CLIP model. For audio features, we employ the AST model.[2] We project the extracted features from all modalities into 256-dimensional vectors using linear layers. We then apply 4 self-attention layers and 1 multi-head cross-attention layer (with 16 heads each) for fusing the candidate music and text features. For contrastive loss, we initialize temperature as 0.1, and set batch size to be 34 per GPU. For training, we use AdamW optimizer [33] with a learning rate of 4e-5 and a decay rate of 5e-4. In the sentence generator module, the maximum sequence length is set to 128, and the temperature is fixed to 0.1. We set the LoRA rank to 32 and train the module for 3 epochs using a learning rate of 5e-4 with linear decay. Training for both modules are conducted

[2]The pretrained weights used are clip-vit-base-patch32 for the CLIP Image encoder and MIT/ast-finetuned-audioset-10-10-0.4593 for the AST.

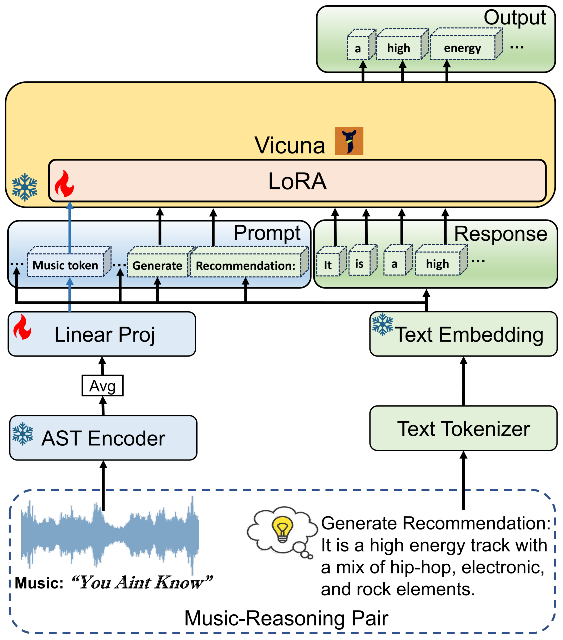


Figure 4. Illustration of sentence generator. During training, we only train the linear projection layer and the additional LoRA weights while keeping the parameters of Vicuna-7B and AST encoder frozen. And the prompt input is "*### Recommender: Music feature: <Music> [music token] </Music>; Generate Recommendation:*". During inference, we give the recommended music title as additional information, and the prompt input is "*### Recommender: Music title: [title]; Music feature: <Music> [music token] </Music>; Generate Recommendation:*". "*[music token]*" corresponds to the embedding vector projected from AST encoder output.

Model	MR ↓	R@1 ↑	R@5 ↑	R@10 ↑	SR ↑
VM-NET [20]	28	5.31	18.14	28.78	-
MVPt [42]	7	20.71	48.89	63.14	-
ViML [36]	5	22.61	52.43	66.56	-
Sum-Fusion	17	10.58	28.47	40.19	21.70
Self-Attn Fusion	5	21.40	50.83	65.29	28.37
Cross-Attn Fusion	3	25.71	56.97	71.07	31.48
MuseChat (ours)	**2**	**32.79**	**63.92**	**76.53**	**40.49**
Chance	250	0.20	1.00	2.00	0.40

Table 1. Music retrieval results for baseline models and MuseChat.

on 16 Nvidia V100 32G GPUs.

5.2. Ranking Evaluation

Following the setup for measuring music retrieval with free-form natural language tasks as described in [36], we randomly create non-overlapping music pools using test data, each containing 500 tracks. Each pool has only one target music track for each video. For the track-level testing, we calculate embeddings for all 12 segments of each 120-second video and music track. We then take the average of these 12 embeddings to create a single representative embedding for each video and each music track. Using these averaged embeddings, we evaluate the performance of the music recommendation module. In the first turn, music is suggested based solely on video features, as we assume that the user does not provide any specific preferences at this point. In the second turn, we include the user's text prompt and candidate music along with the video features. This setup allows us to evaluate the system's ability to modify its initial recommendations based on the new information. For both turns, we rank music tracks by calculating the cosine similarity between the features of the music in the pool and the fusion features. We use various metrics such as Recall@K for K = 1, 5, 10, and Median Rank (MR) to evaluate the performance of the recommendation. We also measure the "success rate" (abbreviated as SR), defined as the percentage of videos whose target music track appears at least once as the top of the recommended list within two turns. Since the MVP model cannot accommodate the user prompt as an input for the second turn, its SR is same as R@1. We report the average performance for each of these metrics across all test music pools.

Baselines Comparison We introduce six baseline models with various structures and modalities to assess the effectiveness of MuseChat in ranking. (1) **VM-NET** [20]. We re-implement it in PyTorch and replace the audio and vision backbone with AST [17] and CLIP [41] respectively for better performance. (2) **MVPt** [42] does not have publicly available source code. Thus, we replace their DeepSim [29] audio encoders with the publicly available AST [17] model. (3) **ViML** [36] also lacks publicly available source code. We replace their DeepSim audio encoders with the publicly available AST model and use their proposed text dropout strategy. Text and video features are fused using Transformer blocks, following the settings described in their paper. (4) **Sum-Fusion** retains the architecture of MuseChat but employs a different fusion mechanism by summing up the vectors of the video, music, and text directly. We use the mean-pooled vector for both candidate music encoded by AST and text features extracted by CLIP text encoder. (5) **Self-Attn Fusion** extends the summation approach by including self-attention layers for text and music modalities before fusion, capturing intra-modal dynamics. (6) **Cross-Attn Fusion** removes self-attention layers and keep the cross-attention layer between music and text branch. We train the baseline models using the same music video pools and loss function as MuseChat. As shown in Table 1, MuseChat outperforms all baseline models. In particular, in R@1 metric, our model achieves a significant gain of **+7.0%** against "Cross-Attn Fusion" and **+10.1%** against the ViML model, showing the effectiveness of our fusion approach as well as the use of all three modalities as inputs. Notably, for models that could take video, candidate music and user prompt as inputs, we report the recall and MR for the second turn to show the effectiveness of our model when dealing with user preferences. The first turn results for these models are detailed in the supplementary materials. Additionally, we explore the contributions of various modalities to the retrieval performance in our model, with these specifics also included in the supplementary materials.

5.3. Modality Ablation Studies

To further show the necessities of combining all three modalities for retrieval, we conduct ablation studies by removing video, candidate music, and text branch in turn during training. When excluding candidate music or text features, we remove the related cross-attention layers, retaining only the self-attention layers, with the cls token used for the summation of modality features. As Table 2 indicates, the model without visual branch, which is only valid for the second turn, exhibits the worst performance. This reveals the significance of visual information. Additionally, when comparing the MVPt [42] model with MuseChat without candidate music, we observe that user prompt helps to retrieve music more accurately.

5.4. Reasoning Evaluation

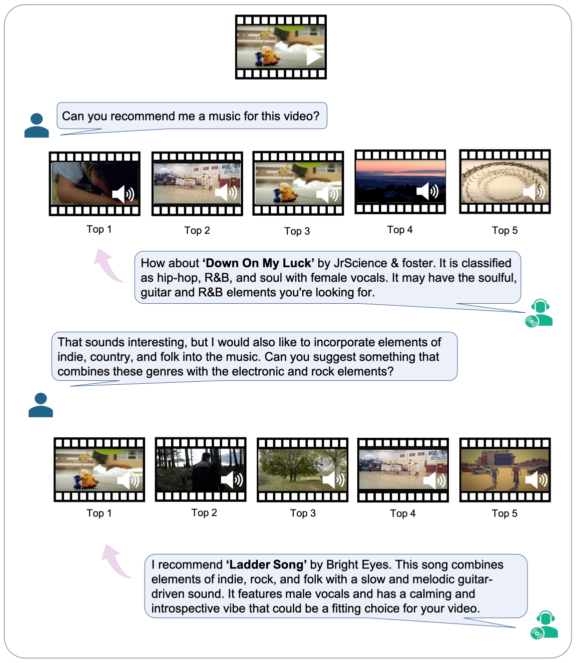
We introduce two baseline models to highlight the significance of training our sentence generator module using both music embeddings and music titles as inputs. The first baseline employs the frozen Vicuna-7B [58] model, which is fine-tuned on the Llama2-7B [45] weights. Since this model cannot process music embeddings, we only present it with the music title. The second baseline (Vicuna w/ Music) uses the same architecture as our sentence generator module but takes only music embeddings as input. We employ various common NLG metrics [6, 56] to evaluate the performance of the models on simulated conversations. As shown in Table 3, the Vicuna-7B model performs the worst. It struggles to capture the musicality of the given track, as it is a text-only model. As for Vicuna w/ Music, while capturing the musicality of the recommended track due to its training on both music and text, it fails to identify the correct music name and artist name based on audio information only. In contrast, by taking both audio information and music title as inputs, our sentence generator module achieves the best

Model	Modality	MR ↓	R@1 ↑	R@5 ↑	R@10 ↑	SR ↑
MVPt [42]	Video → Music	7	20.71	48.89	63.14	20.71
MuseChat w/o Video	(Music, Text) → Music	19	8.12	24.53	37.11	8.12
MuseChat w/o Candidate Music	(Video, Text) → Music	5	22.67	51.53	65.42	26.02
MuseChat (ours)	(Video, Music, Text) → Music	**2**	**32.79**	**63.92**	**76.53**	**40.49**
Chance	-	250	0.20	1.00	2.00	0.40

Table 2. Ablation Studies: Comparing MuseChat's Performance Without Certain Modality Branches and training from scratch.

Model	Input Modality	BertScore [56] (f1) ↑	AB Divergence [6] ↓	$\mathcal{L}_2$ Distance ↓	Fisher-Rao Distance [6] ↓
Vicuna-7B	Music Title	0.9453	3.93	0.382	2.11
Vicuna w/ Music	Music Embeddings	0.9526	2.68	0.279	2.02
MuseChat (ours)	Music Title + Embeddings	**0.9676**	**1.51**	**0.208**	**1.47**

Table 3. Comparison of semantic similarity between output and simulated conversations using various metrics. BERTScore [56] assesses token-level similarity, while AB Divergence, $\mathcal{L}_2$ Distance, and Fisher-Rao Distance are derived based on InfoLM [6].


Can you recommend me a music for this video?

How about **'Down On My Luck'** by JrScience & foster. It is classified as hip-hop, R&B, and soul with female vocals. It may have the soulful, guitar and R&B elements you're looking for.

That sounds interesting, but I would also like to incorporate elements of indie, country, and folk into the music. Can you suggest something that combines these genres with the electronic and rock elements?

I recommend **'Ladder Song'** by Bright Eyes. This song combines elements of indie, rock, and folk with a slow and melodic guitar-driven sound. It features male vocals and has a calming and introspective vibe that could be a fitting choice for your video.

Figure 5. Example: MuseChat retrieves target music in two turns.

performance on reasoning.

We evaluate MuseChat's effectiveness in music recommendations through a qualitative analysis, highlighting its capacity for understanding queries, suggesting relevant music, and generating responses that integrate visual content, user preferences, and audio characteristics. Figure 5 shows how MuseChat interacts with users, dynamically adjusting its recommendations based on a range of contextual cues. We include more examples in the supplementary materials.

We also evaluate our model and two baselines via human assessment, employing a 5-point MOS scale to gauge correctness (music and artist identification), musicality (description of music traits), and clarity (response understandability and coherence). Table 4 shows the human evaluation results. Since Vicuna-7B only takes the music title as input, it achieves a better score in correctness compared to Vicuna w/ Music, which struggles to identify the title and artist of an unknown piece of music based solely on music characteristics. In contrast, the Vicuna w/ Music model captures musicality better than Vicuna-7B, which relies solely on its knowledge base for musicality assessment. MuseChat, however, achieves the best overall performance by both correctly identifying music and artist name and understanding music in the context of video and user preference. Details are included in the supplementary materials.

Model	Correctness	Musicality	Clarity	Overall
Vicuna-7B	3.07	2.54	3.60	3.07
Vicuna w/ Music	1.24	3.50	4.05	2.93
MuseChat (ours)	**4.63**	**4.22**	**4.54**	**4.46**

Table 4. Human evaluation scores for music reasoning outputs.

6. Conclusions

In this work, we establish a closer connection between human and music recommendation system through an interactive dialogue. We build the first conversational music recommendation dataset for videos based on public YouTube-8M dataset. In addition, we propose a system of two modules that interpret multi-modal inputs and deliver its recommendation with reasoning in natural language. Extensive experiments show that our proposed system exhibits strong performance on retrieval task with interpretability and interactivity. Future efforts could be made to design a more integrated system that combines recommendation and reasoning within a single multi-modal LLM.

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. 3

[2] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011. 4

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[4] Jeong Choi, Jongpil Lee, Jiyoung Park, and Juhan Nam. Zero-shot learning for audio-based music classification and tagging. *arXiv preprint arXiv:1907.02670*, 2019. 2

[5] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks. *arXiv preprint arXiv:1606.00298*, 2016. 2

[6] Pierre Colombo, Chloe Clavel, and Pablo Piantanida. Infolm: A new metric to evaluate summarization & data2text generation, 2022. 7, 8

[7] Xing Cui, Zekun Li, Pei Li, Yibo Hu, Hailin Shi, Chunshui Cao, and Zhaofeng He. Chatedit: Towards multi-turn interactive facial image editing via dialogue. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14567–14583, 2023. 3

[8] Yang Deng, Wenxuan Zhang, Weiwen Xu, Wenqiang Lei, Tat-Seng Chua, and Wai Lam. A unified multi-task learning framework for multi-goal conversational recommender systems. *ACM Transactions on Information Systems*, 41(3): 1–25, 2023. 3

[9] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023. 3

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[11] SeungHeon Doh, Minz Won, Keunwoo Choi, and Juhan Nam. Toward universal text-to-music retrieval. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 2

[12] Zhikang Dong and Pawel Polak. Cp-pinns: Changepoints detection in pdes using physics informed neural networks with total-variation penalty. *arXiv preprint arXiv:2208.08626*, 2022. 3

[13] Luke Friedman, Sameer Ahuja, David Allen, Terry Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, Harsh Lara, et al. Leveraging large language models in conversational recommender systems. *arXiv preprint arXiv:2305.07961*, 2023. 3

[14] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2: 100–126, 2021. 3

[15] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 3

[16] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*, 2023. 3

[17] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021. 3, 4, 5, 7

[18] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 976–980. IEEE, 2022. 2

[19] Umut Güçlü, Jordy Thielen, Michael Hanke, and Marcel A. J. van Gerven. Brains on beats, 2016. 4

[20] Sungeun Hong, Woobin Im, and Hyun S Yang. Cbvmr: content-based video-music retrieval using soft intra-modal structure constraint. In *Proceedings of the 2018 ACM on international conference on multimedia retrieval*, pages 353–361, 2018. 6, 7

[21] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 3, 6

[22] Tao Hu, Xuyu Xiang, Jiaohua Qin, and Yun Tan. Audio–text retrieval based on contrastive learning and collaborative attention mechanism. *Multimedia Systems*, pages 1–14, 2023. 2

[23] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel PW Ellis. Mulan: A joint embedding of music audio and natural language. *arXiv preprint arXiv:2208.12415*, 2022. 2, 3

[24] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)*, 54(5):1–36, 2021. 3

[25] Yuming Jiang, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu. Talk-to-edit: Fine-grained facial editing via dialog. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13799–13808, 2021. 3

[26] Xin Jin, Jonathan Larson, Weiwei Yang, and Zhiqiang Lin. Binary code summarization: Benchmarking chatgpt/gpt-4 and other large language models. *arXiv preprint arXiv:2312.09601*, 2023.

[27] Juni Kim, Zhikang Dong, and Pawel Polak. Face-gps: A comprehensive technique for quantifying facial muscle dynamics in videos. *arXiv preprint arXiv:2401.05625*, 2024. 3

[28] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *arXiv preprint arXiv:1703.01789*, 2017. 2

[29] Jongpil Lee, Nicholas J Bryan, Justin Salamon, Zeyu Jin, and Juhan Nam. Disentangled multidimensional metric learning for music similarity. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6–10. IEEE, 2020. 2, 7

[30] Yan-Bo Lin, Yi-Lin Sung, Jie Lei, Mohit Bansal, and Gedas Bertasius. Vision transformers are parameter-efficient audio-visual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2299–2309, 2023. 3

[31] Xiulong Liu, Zhikang Dong, and Peng Zhang. Tackling data bias in music-avqa: Crafting a balanced dataset for unbiased question-answering. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4478–4487, 2024. 3

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[34] Ilaria Manco, Emmanouil Benetos, Elio Quinton, and György Fazekas. Contrastive audio-language learning for music. *arXiv preprint arXiv:2208.12208*, 2022. 2

[35] Ilaria Manco, Benno Weck, Philip Tovstogan, Minz Won, and Dmitry Bogdanov. Song describer: a platform for collecting textual descriptions of music recordings. In *Ismir 2022 Hybrid Conference*, 2022. 2

[36] Daniel McKee, Justin Salamon, Josef Sivic, and Bryan Russell. Language-guided music recommendation for video via prompt analogies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14784–14793, 2023. 2, 6, 7

[37] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5

[38] Jordi Pons and Xavier Serra. musicnn: Pre-trained convolutional neural networks for music audio tagging. *arXiv preprint arXiv:1909.06654*, 2019. 2, 4

[39] Laure Prétet, Gael Richard, and Geoffroy Peeters. Cross-modal music-video recommendation: A study of design choices. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2021. 2

[40] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 3

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 5, 7

[42] Dídac Surís, Carl Vondrick, Bryan Russell, and Justin Salamon. It's time for artistic correspondence in music and video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10564–10574, 2022. 2, 3, 6, 7, 8

[43] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022. 3

[44] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020. 5

[45] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 3, 6, 7

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[47] Xiaolei Wang, Xinyu Tang, Wayne Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. Rethinking the evaluation for conversational recommendation in the era of large language models. *arXiv preprint arXiv:2305.13112*, 2023. 3

[48] Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra. Evaluation of cnn-based automatic music tagging models. *arXiv preprint arXiv:2006.00751*, 2020. 2

[49] Minz Won, Keunwoo Choi, and Xavier Serra. Semi-supervised music tagging transformer. *arXiv preprint arXiv:2111.13457*, 2021. 2

[50] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2clip: Learning robust audio representations from clip. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4563–4567. IEEE, 2022. 2

[51] Bowen Yang, Cong Han, Yu Li, Lei Zuo, and Zhou Yu. Improving conversational recommendation systems' quality with context-aware item meta information. *arXiv preprint arXiv:2112.08140*, 2021. 3

[52] 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*, 2021. 2

[53] 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

[54] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 3

[55] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 3

[56] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020. 7, 8

[57] Hang Zhao, Chen Zhang, Bilei Zhu, Zejun Ma, and Kejun Zhang. S3t: Self-supervised pre-training with swin transformer for music classification. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 606–610. IEEE, 2022. 2

[58] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 6, 7

[59] Yufan Zhou, Ruiyi Zhang, Jiuxiang Gu, Chris Tensmeyer, Tong Yu, Changyou Chen, Jinhui Xu, and Tong Sun. Tigan: Text-based interactive image generation and manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3580–3588, 2022. 3

[60] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 3