

OpenMU: Your Swiss Army Knife for Music Understanding

Anonymous ACL submission

Abstract

We present OpenMU-Bench, a large-scale benchmark suite for addressing the data scarcity issue in training multimodal large language models to understand music. To construct OpenMU-Bench, we used existing datasets and bootstrapped new annotations. OpenMU-Bench also broadens the scope of music understanding by including lyrics understanding and music tool usage. Using OpenMU-Bench, we trained our music understanding model, OpenMU, with extensive ablations, demonstrating that OpenMU outperforms baseline models such as MU-LLaMA. Both OpenMU and OpenMU-Bench are open-sourced to facilitate future research in music understanding and to improve creative music production efficiency.

1 Introduction

Multimodal Large Language Models (MLLMs) have successfully extended large language models (LLMs) by enabling them to perceive, process, and understand data in modalities beyond text (Tsimpoukelli et al., 2021; Liu et al., 2023b; Zhu et al., 2023; McKinzie et al., 2024; Zhang et al., 2023; Gong et al., 2024), such as images, videos, and audio. However, there has been limited effort (Gardner et al., 2024) focused on constructing MLLMs capable of effectively understanding the music modality or addressing Music Information Retrieval (MIR) tasks. MIR is a research field focusing on modeling, understanding and interpreting data relevant to music, aiming to improve the efficiency of music production (Serra et al., 2013). Conventional machine learning algorithms (Wang, 2003; Casey et al., 2008) sparked the success in music searching. Subsequently, deep learning models expanded the success to music tagging (Won et al., 2020), transcription (Gardner et al., 2021; Toyama et al., 2023) and representation learning (Castellon et al., 2021; Li et al., 2024; Won et al., 2024).

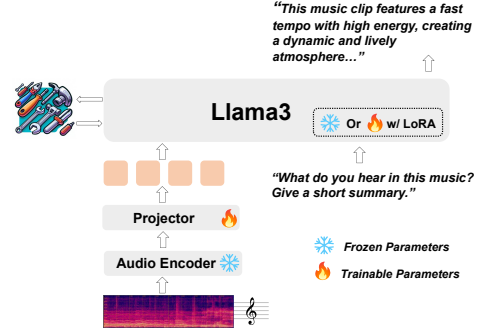


Figure 1: Model architecture of OpenMU. In Stage (1), we only tune the music-language projector. In Stage (2), LoRA adapters are added to the LLM and are tuned together with the projector.

We aim to contribute to the MIR field by training an MLLM, dubbed **OpenMU**, for understanding music clips. Building on the versatile capabilities of LLMs and pretrained audio encoders, OpenMU effectively comprehends and reasons about input music clips, responding with relevant answers accordingly. We also enable OpenMU to utilize well-established music tools familiar to music creative practitioners to encourage synergies between them. OpenMU is expected to greatly improve music production efficiency. Creative practitioners can instruct OpenMU to describe a music clip’s key contents and features, saving minutes of time compared to listening to the full track.

We faced significant challenges when training and evaluating OpenMU, particularly due to the scarcity of data, which is even more pronounced in the music modality (Serra et al., 2013; Seeger, 2003; Holzapfel et al., 2018). To address this issue, we construct **OpenMU-Bench**, a large-scale benchmark for *training and evaluating* MLLMs in music understanding. To construct OpenMU-Bench, we bootstrap new datasets using GPT-3.5, and leverage existing datasets when available. As a result, OpenMU-Bench comprises approximately one million training examples, covering various aspects of music understanding, such as music captioning,

reasoning, multi-choice question answering, lyrics understanding and music tool calling. To the best of our knowledge, no large-scale open-sourced benchmark comparable to OpenMU-Bench currently exists, and we hope it will advance future research and development of music understanding.

In summary, our contributions include: (1) Proposing OpenMU for music understanding. OpenMU is an MLLM dedicated to the music modality, outperforming baseline models such as MU-LLaMA (Liu et al., 2024) in tasks like music captioning, reasoning, and multiple-choice question answering. We carefully evaluate various design choices for OpenMU and provide extensive ablations on key factors. (2) Constructing a large-scale benchmark suite, OpenMU-Bench, consisting of approximately one million music understanding data examples. We bootstrap new data from GPT-3.5 for rich annotations and also leverage existing datasets. (3) Open-sourcing OpenMU and OpenMU-Bench. We hope that they will benefit future research and development in music understanding and enhance creative music production by providing rich resources and consistent evaluations.

2 Related work

Understanding music goes beyond recognizing objective attributes of music such as tempo (Böck et al., 2015; Sun et al., 2021) or instrumentation (Gururani et al., 2019; Zhong et al., 2023). It is also subjective and highly context-dependent, like determining music genres (Kereliuk et al., 2015) or moods (Bogdanov et al., 2019; Koutini et al., 2019). Researchers succeeded in understanding music by classifying music clips into predefined tags (Li et al., 2024; Won et al., 2024). Recently, *music captioning* (Manco et al., 2021) and *reasoning* (Gardner et al., 2024) tasks, where natural language descriptions are employed to describe music clips, have earned increasing attention. Also, the ability of selecting correct answers in *multi-choice question answering* is included in music understanding (Weck et al., 2024). However, there has been limited exploration into enabling MLLMs to utilize external digital tools (i.e., established music tools) for music analysis. We hypothesize that a music understanding model can further improve the workflow of creative practitioners by deeply integrating their familiar music tools. Last but not least, lyrics information processing (Watanabe and Goto, 2020), such as semantic lyrics understand-

ing (Zhang et al., 2022) enhances the understanding of a music clip. Therefore, we integrate it in OpenMU-Bench. Overall, we *broaden the scope of music understanding* by considering two extra aspects beyond music captioning and reasoning: *Music tool using* and *lyrics understanding*.

Foundation models for music understanding. Multimodal LLMs (MLLMs) (Tsimpoukelli et al., 2021; Liu et al., 2023b; Zhu et al., 2023; McKinzie et al., 2024; Gong et al., 2024) fuse non-textual information into LLMs (Liang et al., 2022) to solve real-world tasks requiring the ability of perceiving data in different modalities. The scope of MLLMs is recently expanded to include music. MU-LLaMA (Liu et al., 2024) and MusiLingo (Deng et al., 2024) narrowed down their scope to music captioning and question answering (QA); other critical aspects of music understanding, e.g., key and chord recognition, are not covered. Perhaps the closest to ours is Llark (Gardner et al., 2024). However, neither the model itself nor the music understanding datasets from Llark have been released. None of these models is capable of using music tools, an important ability to interact with creators. In this paper, we propose **OpenMU-Bench** and **OpenMU** to advance the field of music understanding. OpenMU-Bench holistically measures various aspects of music understanding, while OpenMU achieves state-of-art performance on the benchmark. Both OpenMU-Bench and OpenMU are released to facilitate the future research and development in this field.

Music understanding datasets. The proliferation of LLMs has spurred the development of benchmarks designed to holistically measure the genuine capabilities of LLMs. Benchmarks have been designed for NLP tasks (BIG-bench authors, 2023; Hendrycks et al., 2021), and vision-language tasks (Liu et al., 2023c; Fu et al., 2023; Ye et al., 2023). MMMU (Yue et al., 2023) included the music modality into evaluation but at a very narrow scope (334 entries of sheet music). Researchers are striving to address the data scarcity challenge of music: Doh et al. (2023) introduced LP-MusicCaps, associating LLM-augmented captions with music clips from MusicCaps (Agostinelli et al., 2023). Similarly, Liu et al. (2024) developed MusicQA, containing QA and captioning tasks for music clips from MusicCaps, MagnaTagATune (Law et al., 2009a), and MTG-Jamendo (Bogdanov et al., 2019). Concurrently, Deng et al. (2024) proposed MusicInstruct, which targets QA and caption-

ing for clips in MusicCaps. Weck et al. (2024) create MuChoMusic as a music understanding benchmark containing 1,187 multiple-choice questions for evaluation. Building on existing datasets, we construct OpenMU-Bench by additionally bootstrapping new datasets using GPT-3.5. OpenMU-Bench contains about one million examples for training and evaluation across various music understanding tasks. We also standardize evaluation metrics to ensure consistency¹ in reporting results.

3 Constructing OpenMU-Bench

This section outlines the construction of OpenMU-Bench. OpenMU-Bench covers five types of tasks; we explain the dataset construction procedures for each type. In addition to incorporating existing music understanding datasets, we generate new annotations for music clips from datasets that do not contain natural language annotations. Our goal is to integrate as many datasets as possible to enable OpenMU-Bench to comprehensively and systematically evaluate music understanding models. Furthermore, we specify the recommended evaluation metrics to ensure consistent and fair benchmarking. Table 1 shows examples of different OpenMU-Bench task types.

3.1 OpenMU-Bench task types

Music captioning tasks a model with generating textual descriptions capturing musical contents and key features of a music clip. A music understanding model excels at captioning can improve the efficiency of music production by generating music descriptions instantly, eliminating the needs of listening to the entire music track by creators.

Music reasoning (Gardner et al., 2024), tasks the model with answering questions in two aspects. First, it examines the interaction between different musical elements, e.g., how a fast tempo is likely to correspond with a high energy level. Second, it explores how the real-world can interact with the music clip, e.g., how a creator can increase the energy level by using faster tempos (see Table 1).

Tool using. There exists various music technology tools for tasks like tempo estimation, key detection, chord recognition, and instrument iden-

tification². Unlike Llark (Gardner et al., 2024), which aims to address many MIR tasks using only the LLM, OpenMU takes a different view. We aim for OpenMU to integrate and leverage the well-established, rigorously tested MIR tools to solve practical, real-world problems. This approach is motivated by the limitations of current LLMs, which often struggle with tasks like arithmetic (Qin et al., 2023). By combining the strengths of established MIR tools (e.g., tempo estimator) with LLM-based method, OpenMU aims to provide a more robust solution to music understanding.

Lyrics understanding. Lyrics, which carry rich semantic content, are often used to convey moods and emotions (Watanabe and Goto, 2020). We incorporate a lyrics understanding task as a text generation problem into OpenMU-Bench. We consider a model that excels at understanding lyrics to be capable of producing interpretations similar to those of humans, conditioned on the music input.

Multiple-choice questions. Text generation models are inherently difficult to evaluate due to their open-ended nature (Celikyilmaz et al., 2020). This challenge extends to LLM-based music understanding models as well. To ensure holistic evaluations, OpenMU-Bench also assesses these models using multiple-choice questions developed by Weck et al. (2024).

3.2 Individual datasets

As introduced in §2, a few datasets already exist for music understanding. We incorporate these datasets and create new annotations to ensure that OpenMU-Bench has both a large scale and broad coverage. We describe each of the datasets, along with the applied modifications aligning them with the OpenMU-Bench task types. We adhere to existing train/test splits of the datasets when available (c.f. §A.6); Appendix §A.4 details the preprocessing and annotating details of OpenMU-Bench; we highlight only the key information here. Table 5 (§A.1) shows OpenMU-Bench statistics.

MusicCaps, created by Agostinelli et al. (2023), is pivotal for the music captioning task. It contains $\approx 5.5K$ 10-second music clips sourced from AudioSet (Gemmeke et al., 2017), with corresponding gold-standard text captions written by professional musicians. We incorporate MusicCaps into OpenMU-Bench as part of the captioning task. **LP-MusicCaps & LPMusicMTT** (Doh et al., 2023)

¹For example, we found that MU-Llama (Liu et al., 2024) reports BertScore-Recall, while LP-MusicCaps (Doh et al., 2023) reports BertScore-F1. We standardize the metrics when reporting performance on OpenMU-Bench, and hope this paves the way for consistent evaluations of music understanding MLLMs.

²List of MIR software tools: <https://www.ismir.net/resources/software-tools/>

Music Captioning	Question: Write a short description about the musical contents of this clip.
	Answer: The music clip has a tempo similar to a walking pace and belongs to the blues genre. It features the characteristic blues chord progression and soulful melodies often accompanied by instruments typical of the blues genre are also evident in this clip.
Music Reasoning	Question: How would a music producer enhance the high energy of this track?
	Answer: A music producer could enhance the high energy of this track by adding elements like driving basslines, powerful drums, and energetic synthesizers. They could also increase the tempo slightly to create a more intense feel and incorporate dynamic changes in volume and instrumentation to maintain ...
Lyrics Understanding	Question: Please help me to interpret this lyrics: {music_lyrics}
	Answer: Feel like it's about being depressed about your life. You go to work on Monday, see all of the ...
Tool Using	Question: What are the chords between 10 sec and 20 sec of this music?
	Answer: Here are the chords between 10 sec and 20 sec: [GetMusicChords(10, 20)].
Multiple Choice	Question: How would you describe the vocal performance in this piece?
	Options: (A) Soft yet deeply emotional (B) Lamenting (C) Male vocals (D) Operatic
	Answer: (A) Soft yet deeply emotional.

Table 1: Example data entries in OpenMU-Bench.

extend MusicCaps and the MagnaTagATune (Law et al., 2009b) dataset by generating additional textual descriptions. The authors prompt GPT-3.5 to “write”, “summarize”, “paraphrase”, and “predict attributes” new captions. We integrate³ $\approx 8K$ LPMusicCaps and 51K LPMusicMTT training captions into OpenMU-Bench. **MusicInstruct** (Deng et al., 2024) also extends MusicCaps by creating question-answer pairs for the MusicCaps clips using GPT-4. This dataset contains $\approx 60K$ question-answer pairs, which are categorized into two versions: a short version (**MI-short**) focusing on musical content such as tempo and genre, and a long version (**MI-long**) that paraphrases the MusicCaps captions. We integrate MusicInstruct into OpenMU-Bench as a captioning task, and report performance on both versions separately.

MusicQA, developed by Liu et al. (2024) via prompting MPT (MosaicML-NLP-Team, 2023), is employed to train their MU-LLaMA. MusicQA consists of MusicCaps clips for pretraining, MagnaTagATune (Law et al., 2009b) clips for finetuning, and MTG-Jamendo (Bogdanov et al., 2019) clips for testing. We incorporate MusicQA-Finetune and MusicQA-Test into OpenMU-Bench, while MusicQA-Pretrain, which contains the test split of MusicCaps, is excluded to prevent potential train-test leakage (Deng et al., 2024). Following Liu et al. (2024), we separate MusicQA into captioning and reasoning parts.

Music4all, developed by Pegoraro Santana et al. (2020), consists of $\approx 100K$ music clips with rich metadata, including attributes like energy, valence, and genre. Based on this metadata, we prompt

GPT-3.5 to generate annotations for both the captioning and reasoning tasks. The prompts used for these annotations are provided in the Appendix §A.5. **GTZAN**, developed by Tzanetakis and Cook (2002), contains $\approx 1K$ 30-second music clips, each labeled with genre tags and we create extra tempo tags with Madmom (Böck et al., 2016). Based on these tags, we generate captioning and reasoning annotations with prompting. **MusicNet** (Thickstun et al., 2017) contains 1 million dense annotations at precise timestamps for 330 classical music recordings. The annotations are of high quality, but primarily focus on instruments. As a result, we integrate MusicNet into OpenMU-Bench as part of the captioning task, retaining only annotations that span three seconds or longer. **MagnaTagATune (MTT)** has been included in OpenMU-Bench as part of the captioning task, thanks to the annotations by Doh et al. (2023). Given its significance in the MIR community (Won et al., 2020), we also create an additional 90K reasoning annotations for training and testing. **MTG-Jamendo** (Bogdanov et al., 2019) consists of $\approx 55K$ full music tracks, each tagged with genre, instrument, and mood. We randomly select 30-second music clips⁴ and generate annotations for captioning and reasoning tasks by prompting GPT-3.5.

Tool using. To the best of our knowledge, there is no existing dataset designed to train MLLMs to use MIR tools. We thus generate training and testing datasets for solving four MIR tasks with tools: chord recognition, tempo estimation, key detection, and downbeat extraction. We demonstrate that OpenMU quickly learns to utilize these tools to answer queries related to MIR information. We implemented these tools by wrapping the Python

³We do not use the “attribute prediction” annotations, following the recommendation from the LPMusic authors: <https://huggingface.co/datasets/seungheondoh/LP-MusicCaps-MC>

⁴We provide scripts for extracting music clips.

package Madmom (Böck et al., 2016); §A.7 shows implementation details.

For **Lyrics understanding**, we integrate Zhang et al. (2022)’s annotations, containing internet interpretations to the lyrics of Music4all music clips. For **Multiple-choice questions**, we integrate MuChMusic (Weck et al., 2024) for evaluation. The task involves answering questions about music knowledge and reasoning by selecting the correct option from four provided choices.

3.3 Evaluation metrics

OpenMU-Bench leverages common evaluation metrics for text generation tasks: captioning, reasoning, and lyrics understanding. BLEU-1, BLEU (Papineni et al., 2002)⁵, Meteor (Banerjee and Lavie, 2005), Rouge-1, and Rouge-L (Lin, 2004) measure an answer’s textual overlap with the gold standard, while BertScore (Zhang et al., 2020) measures similarity in the semantic representation space of BERT. For all evaluations, we report the scores computed using the F-measure. We report accuracy for the task of multiple-choice questions.

4 OpenMU

4.1 Model architecture

Encoding music clips. We use AudioMAE (Huang et al., 2022) to encode an input music clip. Specifically, we use the “ViT-B AS-2M pretrained + fine-tuned” version of AudioMAE, which is a Vision Transformer (Dosovitskiy et al., 2021) initially pre-trained with a masked auto-encoding reconstruction loss (He et al., 2022), followed by finetuning on tagging tasks (Gemmeke et al., 2017), both on AudioSet2M (Gemmeke et al., 2017). The choice of using AudioMAE over other music encoders, such as MERT (Li et al., 2024) or Jukebox-5B (Dhariwal et al., 2020; Castellon et al., 2021), is motivated by two primary reasons. First, more than half of the audio clips in AudioSet2M⁶ consist of music or musical instrument recordings, resulting in approximately 3,137 hours of music data (larger than the 910-hour data used to train MERT-95M-public (Li et al., 2024)). Audio encoders pretrained on AudioSet have shown competitive performance in music tagging tasks (Koutini et al., 2021; Nizumi et al., 2022). Second, the size of the multimodal encoder is not a performance bottleneck

⁵Following the machine translation literature, our BLEU refers to BLEU-4.

⁶AudioSet2M Ontology: <https://research.google.com/audioset/>

(McKinzie et al., 2024). Instead, the smaller number of parameters in ViT-B (86M) facilitates more efficient training.

LLM. We use the open-sourced 1B and 8B models from the Llama3 family (Dubey et al., 2024) as our LLM. Compared to previous Llama models (Touvron et al., 2023), Llama3 has been trained on higher-quality datasets and at larger scales, achieving great performance (Achiam et al., 2023) on numerous tasks.

Music-language projector links the representation space of the music encoder with the LLM. Studies (McKinzie et al., 2024; Liu et al., 2023a) have shown that the architecture of the projector itself has little impact on downstream task performance; the number of tokens from the multimodal encoder is significantly important. We use a two-layer MLP with GELU non-linearity (Hendrycks and Gimpel, 2017) and evaluate the effect of varying the number of music tokens in §5.1.

Overall, OpenMU follows the well-tested architecture of MLLMs (McKinzie et al., 2024) as shown in Figure 1. In contrast to previous music MLLMs such as MU-LLaMA, OpenMU is also capable of interacting with external MIR tools such as tempo estimator (Böck et al., 2016).

4.2 Training details

Dataset preprocessing. When processing the music clips, we limit their maximum length to 30 seconds and zero-pad those shorter than 30 seconds. Music clips are resampled to 16 kHz and then converted to a 128-bin Mel-spectrogram with a 25-ms Hanning window and 10-ms hop size. Consequently, each music clip is converted to a Mel-spectrogram of shape (3072, 128). As AudioMAE encodes 10-second inputs, we segment each Mel-spectrogram into three parts, encode them separately, and then concatenate them. As a result, each 30-second music clip is encoded by 1536 tokens, and each of them has a shape of (1, 768).

Throughout our experiments, we used between 8 and 16 A100 40GB GPUs, depending on the experimental setup (c.f. §5). In all experiments, we set the maximum context length of the LLM to 2048 tokens. We utilized DeepSpeed ZeRO-3 (Rajbhandari et al., 2020) and FlashAttention2 (Dao et al., 2022) to enable fast and efficient training. It took approximately three days to train OpenMU on the captioning and reasoning subsets of OpenMU-Bench (around 1 million data examples).

Training setup of OpenMU-Bench largely fol-

lows the common practice of MLLM training (Yin et al., 2023; Liu et al., 2023b; McKinzie et al., 2024), consisting of:

Stage (1) Captioning. We train OpenMU to generate captions, conditioned solely on the input music clip. The goal of Stage (1) training is to align the representation spaces of AudioMAE and LLM, with the only trainable module in this stage being the music-language projector. We use the captioning subset of OpenMU-Bench in this stage. A key configuration is the number of music tokens fed into the LLM; we discuss it in detail in §5.1. The remaining hyperparameters largely follow Liu et al. (2023b) shown in Appendix §A.2.

Stage (2) Instruction Tuning. After aligning the music and text representation spaces in Stage (1), Stage (2) trains OpenMU to follow various instructions about music, such as inferring music genres or reasoning about the content of a music clip. We add LoRA adapters (Hu et al., 2022) to OpenMU’s LLM, and then finetune the model on OpenMU-Bench’s captioning and reasoning tasks. We focus on two critical research questions in this stage. First, we ablate OpenMU’s task performance with respect to its LoRA parameters (see §5.2). Second, we investigate in-depth OpenMU’s use of music information. Given the large-scale pretraining data of LLM, we hypothesize that OpenMU might be able to make correct predictions for knowledge-intensive questions even without relying on musical information within a music clip. We test this hypothesis and show that in order to achieve higher performance, OpenMU indeed relies on information from the music clip, demonstrating its genuine ability to understand music.

5 Training OpenMU

In this section, we analyze and discuss key factors when training OpenMU. Throughout our discussion, we use the 8B Llama3 model as the LLM.

5.1 Number of music tokens

McKinzie et al. (2024) show that the number of image tokens is more significant than the architecture of the vision-language projector when training vision-language models. To the best of our knowledge, no prior research has explored this aspect when training foundation models for understanding music. This is particularly important as *music clips can often be lengthy, leading to a large number of music tokens*. E.g., AudioMAE encoder outputs

1536 tokens for representing a 30-second music clip (c.f. §4.2). Using all the music tokens ensures the maximum utilization of information in the music modality, but it hinders training efficiency and consumes a large portion of the context window allowed by the LLM (2048 in our case). We show the effects of the number of music tokens in §5.3.

5.2 LoRA parameters

Low-Rank Adaptation (LoRA; Hu et al. (2022)), is employed in Stage (2) training to efficiently adapt OpenMU’s LLM to follow instructions in the music domain. For an LLM weight parameter matrix $W \in \mathbb{R}^{d \times k}$, instead of directly modifying W , LoRA introduces and trains two matrices, $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ for adapting W to a downstream task: $W \leftarrow W + \frac{\alpha}{r}BA$.

W remains unchanged in training; LoRA rank r determines the number of trainable parameters by controlling the size of B and A . BA represents the parameter changes needed to adapt to a downstream task, scaled by $\frac{\alpha}{r}$. α is a hyperparameter, and typically $r < \alpha$. For OpenMU, we fix $\alpha = 128$ following Liu et al. (2023b,a), while varying the value of r . A smaller rank r imposes a stricter bottleneck on B and A , requiring the learned parameter differences, represented by BA , to rely on fewer trainable parameters to capture concise and genuine information about the downstream task, which are subsequently scaled by a larger $\frac{\alpha}{r}$. In contrast, a larger r introduces more trainable parameters, which may be prone to learning shortcuts, redundant information, or noise (Geirhos et al., 2020) during adaptation to the downstream task, subsequently scaled by a smaller $\frac{\alpha}{r}$. §5.3 shows the effectiveness of LoRA parameters.

5.3 Results and ablations

OpenMU-Bench task performance. Figure 2 shows the performance of OpenMU variants on the reasoning task. Appendix Figure 5 shows the captioning results. For each evaluation, we report the macro-average performance of each OpenMU variant across all subtasks in OpenMU-Bench. Additionally, Appendix Figure 6 displays the evaluation results using BertScore (Zhang et al., 2020) as the metric.

Several observations can be made. First, the number of music tokens plays a critical role in task performance, echoing the conclusion drawn by McKinzie et al. (2024). Mean-pooling every 8 tokens shows clear advantages over 32 and 128

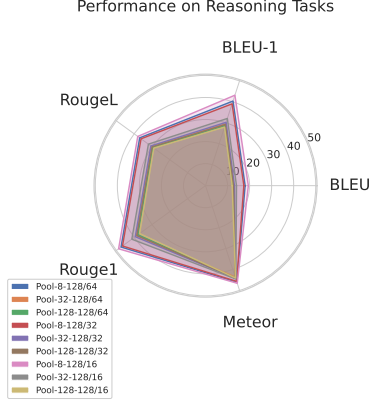


Figure 2: OpenMU variants performance on OpenMU-Bench reasoning tasks. For each evaluation metric, such as BLEU, we report the macro average of the model’s performance across all OpenMU-Bench sub-tasks. “Pool-8-128/16” represents mean pooling every eight music tokens and using LoRA parameter $\frac{128}{16}$.

tokens, likely due to its preservation of music information. However, mean-pooling 32 tokens offers only limited improvement over 128 tokens, and the performance decline appears to plateau. Please refer to Figure 4 (Appendix §A.2) for detailed learning curves. It is likely that crucial music information is already lost when mean-pooling 32 tokens. Second, the effectiveness of LoRA parameters show limited impacts on task performance, similar to the findings in Gong et al. (2024). As a result, we will focus on the model variant with mean-pooling 8 tokens, and LoRA parameters 128/16 in the next sections.

Music information utility. The LLM’s pretraining data likely already contains extensive knowledge about music, enabling OpenMU to answer music-related questions without relying on the given music input. We test a key hypothesis: Does OpenMU genuinely utilize information from the music input to answer questions, or is it merely hallucinating? We evaluate OpenMU variants on MuChoMusic (Weck et al., 2024), a dataset containing multiple-choice questions about music understanding. Questions such as “Which sub-genre of rock music would best classify this piece?” require the model to select the correct option from four candidates. Notably, such questions could be answered based on the most common or probable sub-genre from the LLM’s pretraining data, allowing the model to perform reasonably well without actually relying on the music input. Figure 3 presents the MuChoMusic results of OpenMU variants. “No Music” refers to replacing the input music clip with

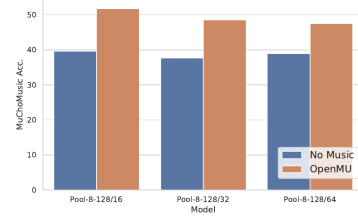


Figure 3: OpenMU performance on MuChoMusic.

a white noise clip, while “OpenMU” displays the results when actual music clips are used. It is evident that music information is crucial for OpenMU to achieve strong performance; OpenMU effectively utilizes music information rather than merely relying on its internal knowledge.

6 Overall results

In this section, we compare OpenMU with other publically available models such as MU-LLaMA (Liu et al., 2024) and Qwen-Audio (Chu et al., 2023) on OpenMU-Bench. For OpenMU, we use the variant of mean-pooling 8 tokens with LoRA parameters 128/16.

Music captioning and reasoning results are presented in Table 2. We observe that OpenMU outperforms MU-LLaMA and Qwen-Audio across various captioning and reasoning tasks. Interestingly, MU-LLaMA lags behind OpenMU on MusicCaps, despite the fact that the MusicCaps test set was used during MU-LLaMA’s pretraining stage (Liu et al., 2024; Deng et al., 2024). We believe this is due to the small size of MusicCaps—its effectiveness was likely overshadowed by the larger finetuning datasets used for MU-LLaMA.

MU-LLaMA outperforms OpenMU and Qwen-Audio on the MusicInstruct-short (MI-short) captioning task (Deng et al., 2024) and the MusicQA-test reasoning task (Liu et al., 2024) in terms of surface form matching metrics by a large margin. However, we found that the gold references in these two subsets are biased to contain a large portion of repeated parts from the questions. For example, in MusicQA-test reasoning, to the question “What is the alternative genre of music in the audio?”, the gold standard reference is “The alternative genre of music in the audio is postrock.”

This observation is further supported by objective metrics. Compared to the edit distance of 225 and the Jaccard similarity score of 23.9% for MTT, MusicQA-test reasoning has an edit distance of 90 and a Jaccard similarity score of 36.2%. Since MU-LLaMA tends to repeat the question before

	BLEU-1			BLEU			Rouge1			RougeL			BertScore			Meteor		
	OMU	MUL	Qwen	OMU	MUL	Qwen	OMU	MUL	Qwen	OMU	MUL	Qwen	OMU	MUL	Qwen	OMU	MUL	Qwen
MusicCaps	25.62	9.78	26.16	2.89	0.73	4.74	27.99	21.22	26.98	18.56	15.64	19.83	86.63	86.85	87.15	21.83	11.03	19.30
MI-short	18.92	50.01	35.70	8.75	23.90	16.00	41.03	52.69	46.97	37.75	47.73	41.85	89.90	92.83	91.82	43.81	49.10	45.14
MI-long	36.66	2.13	9.36	4.18	0.18	0.73	38.74	19.34	22.55	21.70	13.55	15.17	87.24	85.90	85.65	24.43	8.83	11.54
LPMusicMTT	23.83	18.87	11.49	2.56	0.56	0.28	29.23	21.68	16.93	21.57	16.07	13.50	89.75	88.37	86.33	24.60	13.96	12.54
Music4all*	51.03	5.11	12.06	18.69	0.36	0.58	51.31	19.51	17.55	34.80	14.31	12.80	91.04	86.64	83.96	43.58	10.07	10.27
MusicQA-Test	19.60	19.65	15.64	2.22	5.08	1.40	23.96	31.74	18.72	17.39	28.07	15.15	87.30	89.45	85.57	26.34	21.73	13.53
GTZAN*	45.38	4.56	12.22	11.78	0.32	0.50	44.01	20.41	17.04	27.62	15.17	12.42	89.34	87.03	84.16	34.71	10.65	9.63
MusicNet*	52.68	1.17	4.83	22.14	0.00	0.18	56.44	14.49	12.60	38.11	12.25	10.71	91.97	85.59	83.94	45.92	6.56	6.34
MTG-Jam.*	47.56	5.32	12.10	15.83	0.41	0.67	49.66	20.29	16.30	33.76	15.94	12.64	90.79	87.72	85.20	39.44	10.06	9.53

	BLEU-1			BLEU			Rouge1			RougeL			BertScore			Meteor		
	OMU	MUL	Qwen	OMU	MUL	Qwen	OMU	MUL	Qwen	OMU	MUL	Qwen	OMU	MUL	Qwen	OMU	MUL	Qwen
Music4all*	49.20	18.13	21.3	23.31	5.97	6.53	53.26	34.11	31.66	41.08	25.01	23.32	92.34	89.63	88.25	49.96	22.51	31.31
MusicQA-Test	24.84	40.64	27.6	9.46	22.47	12.87	35.86	51.29	45.20	30.66	47.54	40.93	89.70	92.59	90.93	40.04	46.15	43.35
GTZAN*	50.26	16.16	22.61	22.07	5.95	7.13	52.96	35.18	33.17	38.57	26.20	24.09	92.02	89.89	88.28	46.87	21.84	31.64
MTT*	45.52	21.70	25.18	21.18	8.31	8.30	50.83	38.70	33.35	39.93	29.92	25.64	92.03	90.63	88.93	48.06	26.33	29.40
MTG-Jam.*	45.87	23.69	21.82	21.12	8.59	6.96	50.78	38.45	32.19	39.74	29.27	24.57	92.01	90.47	88.52	47.97	26.42	30.70

Table 2: OpenMU-Bench captioning (top) and reasoning (bottom) results (in %) of OpenMU (OMU), MU-LLaMA (MUL), and Qwen-Audio (Qwen). *: datasets with our new annotations.

	Accuracy			IFR
	All	Knowledge	Reasoning	All
MusiLingo	21.1	22.0	19.2	71.6
MU-LLaMA	32.4	32.3	31.3	79.4
M2UGen	42.9	44.9	41.2	96.4
OpenMU	51.8	51.4	51.4	94.8
Random	25.0	25.0	25.0	100.0

Table 3: MuChoMusic accuracy and instruction-following rate (IFR) of OpenMU and prior music understanding models. Numbers are in %. MuChoMusic contains multiple-choice questions; MusiLingo, MU-LLaMA, M2UGen performances are from Weck et al. (2024). “Random” shows random guessing results. We assume “Random” will always select an option, hence its IFR is 100%.

providing an answer, the behavior might have inflated the surface-level form matching scores in these subsets. As a result, we recommend practitioners downweight these subsets when evaluating music understanding models.

	BLEU-1	BLEU	Rouge1	RougeL	BertScore	Meteor
BART-Fu.	25.79	6.48	32.18	17.99	83.03	27.97
OpenMU	25.60	5.19	31.31	17.03	83.14	27.01

Table 4: Lyrics understanding results (in %).

Multiple-choice questions. We compare OpenMU with MU-LLaMA, along with other available music understanding models, on the multiple-choice question dataset MuChoMusic. MusiLingo (Deng et al., 2024) is a concurrent work to MU-LLaMA while M2UGen (Hussain et al., 2023) adds music generation ability to MU-LLaMA. Table 3 shows that OpenMU achieves state-of-the-art music understanding performance on MuChoMusic.

Lyrics understanding results of OpenMU and BART-fusion (Zhang et al., 2022), a model specifically designed for lyrics understanding, are shown in Table 4. For simplicity, we reuse the same hy-

perparameters from Stage (2) training, except for extending the training to 20 epochs. OpenMU outperforms BART-fusion in BertScore but slightly lags behind on other metrics. Future models could explore further hyperparameter tuning or architectural modifications to improve performance. **Tool using accuracy.** When calling external MIR tools, OpenMU achieves 94.95% in chords identification, 95.83% in tempo estimation, 100% in key and downbeats estimation. We consider an exact match as a hit. For example, in chords estimation, if the gold reference is “[GetMusicChords(10, 20)]”, the model must accurately output the type and arguments of the tool call to be considered a hit. Extra calls are considered a miss. OpenMU performs well on this task and learns to call MIR tools effectively. It is promising to integrate more MIR tools to handle a broader range of task types and complexities.

7 Conclusion

We presented OpenMU-Bench, a large-scale benchmark suite containing approximately one million examples for training and evaluating music understanding models. We trained OpenMU, with extensive ablations and demonstrated that it outperforms two baseline models. Both OpenMU and OpenMU-Bench are open-sourced to facilitate future research in music understanding and enhance the efficiency of creative music production. Future work may explore extending OpenMU to support multiple music clips as input and enable in-context learning. Integrating more MIR tools, combining the strengths of LLMs and established tools for deeper music understanding, are also promising.

Limitations

When constructing OpenMU-Bench, we incorporated existing music understanding datasets and leveraged GPT-3.5 to bootstrap new annotations. However, hiring professional musicians for manual annotations would be prohibitively expensive and challenging due to their scarcity. Our approach prioritizes large-scale dataset construction, and future work can incorporate expert-verified subsets to enhance quality.

Following the current music understanding literature, OpenMU-Bench considers five subtasks, as shown in Table 1. However, this list is not exhaustive, because real-world music analysis encompasses wider range of tasks, such as phonetic analysis of music. Future work can incorporate more tasks, expanding the scope of OpenMU-Bench.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. 2023. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*.

Satanjeev Banerjee and Alon Lavie. 2005. **METEOR: An automatic metric for MT evaluation with improved correlation with human judgments**. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

BIG-bench authors. 2023. **Beyond the imitation game: Quantifying and extrapolating the capabilities of language models**. *Transactions on Machine Learning Research*.

Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. 2016. **Madmom: a new Python Audio and Music Signal Processing Library**. In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 1174–1178, Amsterdam, The Netherlands.

Sebastian Böck, Florian Krebs, and Gerhard Widmer. 2015. Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In *ISMIR*, pages 625–631.

Dmitry Bogdanov, Minz Won, Philip Tovstogan, Alistair Porter, and Xavier Serra. 2019. **The mtg-jamendo dataset for automatic music tagging**. In *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, United States.

Michael A. Casey, Remco Veltkamp, Masataka Goto, Marc Leman, Christophe Rhodes, and Malcolm Slaney. 2008. **Content-based music information retrieval: Current directions and future challenges**. *Proceedings of the IEEE*, 96(4):668–696.

Rodrigo Castellon, Chris Donahue, and Percy Liang. 2021. Codified audio language modeling learns useful representations for music information retrieval. *arXiv preprint arXiv:2107.05677*.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Zihao Deng, Yinghao Ma, Yudong Liu, Rongchen Guo, Ge Zhang, Wenhua Chen, Wenhao Huang, and Emmanuel Benetos. 2024. **MusiLingo: Bridging music and text with pre-trained language models for music captioning and query response**. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3643–3655, Mexico City, Mexico. Association for Computational Linguistics.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. 2020. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*.

SeungHeon Doh, Keunwoo Choi, Jongpil Lee, and Juhan Nam. 2023. **Lp-musiccaps: Llm-based pseudo music captioning**. In *Proceedings of the 24th International Society for Music Information Retrieval Conference*, pages 409–416. ISMIR.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela

764	Fan, et al. 2024. The llama 3 herd of models. <i>arXiv preprint arXiv:2407.21783</i> .	Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-	816
765		Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu	817
		Chen. 2022. LoRA: Low-rank adaptation of large	818
766	Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin,	language models . In <i>International Conference on</i>	819
767	Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng,	<i>Learning Representations</i> .	820
768	Ke Li, Xing Sun, et al. 2023. Mme: A comprehensive		
769	evaluation benchmark for multimodal large language	Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski,	821
770	models. <i>arXiv preprint arXiv:2306.13394</i> .	Michael Auli, Wojciech Galuba, Florian Metze, and	822
		Christoph Feichtenhofer. 2022. Masked autoen-	823
		coders that listen. <i>Advances in Neural Information</i>	824
771	Josh Gardner, Simon Durand, Daniel Stoller, and Rachel	<i>Processing Systems</i> , 35:28708–28720.	825
772	Bittner. 2024. Llark: A multimodal instruction-		
773	following language model for music. <i>Proc. of</i>	Atin Sakkeer Hussain, Shansong Liu, Chenshuo Sun,	826
774	<i>the International Conference on Machine Learning</i>	and Ying Shan. 2023. M2ugen: Multi-modal music	827
775	(ICML).	understanding and generation with the power of large	828
		language models. <i>arXiv preprint arXiv:2311.11255</i> .	829
776	Josh Gardner, Ian Simon, Ethan Manilow, Curtis		
777	Hawthorne, and Jesse Engel. 2021. Mt3: Multi-	Corey Kereliuk, Bob L Sturm, and Jan Larsen. 2015.	830
778	task multitrack music transcription. <i>arXiv preprint</i>	Deep learning and music adversaries. <i>IEEE Transac-</i>	831
779	<i>arXiv:2111.03017</i> .	<i>tions on Multimedia</i> , 17(11):2059–2071.	832
780	Robert Geirhos, Jörn-Henrik Jacobsen, Claudio	Khaled Koutini, Shreyan Chowdhury, Verena	833
781	Michaelis, Richard Zemel, Wieland Brendel,	Haunschmid, Hamid Eghbal-Zadeh, and Gerhard	834
782	Matthias Bethge, and Felix A Wichmann. 2020.	Widmer. 2019. Emotion and theme recognition in	835
783	Shortcut learning in deep neural networks. <i>Nature</i>	music with frequency-aware rf-regularized cnns.	836
784	<i>Machine Intelligence</i> , 2(11):665–673.	<i>arXiv preprint arXiv:1911.05833</i> .	837
785	Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman,	Khaled Koutini, Jan Schlüter, Hamid Eghbal-Zadeh,	838
786	Aren Jansen, Wade Lawrence, R Channing Moore,	and Gerhard Widmer. 2021. Efficient training of	839
787	Manoj Plakal, and Marvin Ritter. 2017. Audio set:	audio transformers with patchout. <i>arXiv preprint</i>	840
788	An ontology and human-labeled dataset for audio	<i>arXiv:2110.05069</i> .	841
789	events. In <i>2017 IEEE international conference on</i>		
790	<i>acoustics, speech and signal processing (ICASSP)</i> ,	Edith Law, Kris West, Michael I Mandel, Mert Bay, and	842
791	pages 776–780. IEEE.	J Stephen Downie. 2009a. Evaluation of algorithms	843
		using games: The case of music tagging. In <i>ISMIR</i> ,	844
		pages 387–392. Citeseer.	845
792	Yuan Gong, Hongyin Luo, Alexander H. Liu, Leonid		
793	Karlsinsky, and James R. Glass. 2024. Listen, think,	Edith Law, Kris West, Michael I. Mandel, Mert Bay, and	846
794	and understand . In <i>The Twelfth International Confer-</i>	J. Stephen Downie. 2009b. Evaluation of algorithms	847
795	<i>ence on Learning Representations</i> .	using games: The case of music tagging . In <i>ISMIR</i> ,	848
		pages 387–392.	849
796	Siddharth Gururani, Mohit Sharma, and Alexander	Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran	850
797	Lerch. 2019. An attention mechanism for musical	Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin,	851
798	instrument recognition. In <i>ISMIR</i> , pages 83–90.	Anton Ragni, Emmanouil Benetos, Norbert Gyenge,	852
		Roger Dannenberg, Ruibo Liu, Wenhao Chen, Gus	853
799	Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Pi-	Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike	854
800	otr Dollár, and Ross Girshick. 2022. Masked autoen-	Guo, and Jie Fu. 2024. MERT: Acoustic music un-	855
801	coders are scalable vision learners. In <i>Proceedings</i>	derstanding model with large-scale self-supervised	856
802	<i>of the IEEE/CVF conference on computer vision and</i>	training . In <i>The Twelfth International Conference on</i>	857
803	<i>pattern recognition</i> , pages 16000–16009.	<i>Learning Representations</i> .	858
804	Dan Hendrycks, Collin Burns, Steven Basart, Andy	Sheng Liang, Mengjie Zhao, and Hinrich Schuetze.	859
805	Zou, Mantas Mazeika, Dawn Song, and Jacob Stein-	2022. Modular and parameter-efficient multimodal	860
806	hardt. 2021. Measuring massive multitask language	fusion with prompting . In <i>Findings of the Associa-</i>	861
807	understanding. <i>Proceedings of the International Con-</i>	<i>tion for Computational Linguistics: ACL 2022</i> , pages	862
808	<i>ference on Learning Representations (ICLR)</i> .	2976–2985, Dublin, Ireland. Association for Compu-	863
		tational Linguistics.	864
809	Dan Hendrycks and Kevin Gimpel. 2017. Bridging non-	Chin-Yew Lin. 2004. ROUGE: A package for auto-	865
810	linearities and stochastic regularizers with gaussian	matic evaluation of summaries . In <i>Text Summariza-</i>	866
811	error linear units .	<i>tion Branches Out</i> , pages 74–81, Barcelona, Spain.	867
		Association for Computational Linguistics.	868
812	Andre Holzapfel, Bob Sturm, and Mark Coeckelbergh.		
813	2018. Ethical dimensions of music information re-	Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae	869
814	trieval technology. <i>Transactions of the International</i>	Lee. 2023a. Improved baselines with visual instruc-	870
815	<i>Society for Music Information Retrieval</i> , 1(1):44–55.	tion tuning.	871

872	Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In <i>NeurIPS</i> .	929
873		930
874	Shansong Liu, Atin Sakkeer Hussain, Chenshuo Sun, and Ying Shan. 2024. Music understanding llama: Advancing text-to-music generation with question answering and captioning. In <i>ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)</i> , pages 286–290. IEEE.	931
875		932
876		
877		933
878		934
879		935
880		936
881	Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2023c. Mmbench: Is your multi-modal model an all-around player? <i>arXiv preprint arXiv:2307.06281</i> .	937
882		938
883		
884		939
885		940
886	Ilaria Manco, Emmanouil Benetos, Elio Quenton, and György Fazekas. 2021. Muscaps: Generating captions for music audio. In <i>2021 International Joint Conference on Neural Networks (IJCNN)</i> , pages 1–8. IEEE.	941
887		942
888		943
889		944
890		945
891	Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. 2024. Mm1: Methods, analysis & insights from multimodal llm pre-training. <i>arXiv preprint arXiv:2403.09611</i> .	946
892		947
893		948
894		
895		949
896		950
897	MosaicML-NLP-Team. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. Accessed: 2023-05-05.	951
898		
899		952
900	Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. 2022. Masked spectrogram modeling using masked autoencoders for learning general-purpose audio representation. In <i>HEAR: Holistic Evaluation of Audio Representations</i> , pages 1–24. PMLR.	953
901		954
902		955
903		956
904		
905		957
906	Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. <i>Bleu: a method for automatic evaluation of machine translation</i> . In <i>Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics</i> , pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.	958
907		959
908		
909		960
910		961
911		962
912		963
913	Igor André Pegoraro Santana, Fabio Pinhelli, Juliano Donini, Leonardo Catharin, Rafael Biazus Mangolin, Yandre Maldonado e Gomes da Costa, Valéria Delisandra Feltrim, and Marcos Aurélio Domingues. 2020. <i>Music4all: A new music database and its applications</i> . In <i>2020 International Conference on Systems, Signals and Image Processing (IWSSIP)</i> , pages 399–404.	964
914		965
915		966
916		967
917		968
918		969
919		
920		970
921	Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, Yi Ren Fung, Yusheng Su, Huadong Wang, Cheng Qian, Runchu Tian, Kunlun Zhu, Shi Liang, Xingyu Shen, Bokai Xu, Zhen Zhang, Yining Ye, Bo Li, Ziwei Tang, Jing Yi, Yu Zhu, Zhenning Dai, Lan Yan, Xin Cong, Ya-Ting Lu, Weilin Zhao, Yuxiang Huang, Jun-Han Yan, Xu Han, Xian	971
922		972
923		973
924		974
925		
926		975
927		976
928		977
		978
		979
		980
		981
		982
		983

Avery Wang. 2003. [An industrial strength audio search algorithm](#). In *International Society for Music Information Retrieval Conference*.

Kento Watanabe and Masataka Goto. 2020. [Lyrics information processing: Analysis, generation, and applications](#). In *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, pages 6–12. Online. Association for Computational Linguistics.

Benno Weck, Ilaria Manco, Emmanouil Benetos, Elio Quinton, György Fazekas, and Dmitry Bogdanov. 2024. Muchomusic: Evaluating music understanding in multimodal audio-language models. In *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*.

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

Minz Won, Yun-Ning Hung, and Duc Le. 2024. A foundation model for music informatics. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1226–1230. IEEE.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*.

Ruibin Yuan, Yinghao Ma, Yizhi Li, Ge Zhang, Xingran Chen, Hanzhi Yin, Le Zhuo, Yiqi Liu, Jiawen Huang, Zeyue Tian, et al. 2023. Marble: Music audio representation benchmark for universal evaluation. *arXiv preprint arXiv:2306.10548*.

X Yue, Y Ni, K Zhang, T Zheng, R Liu, G Zhang, S Stevens, D Jiang, W Ren, Y Sun, et al. 2023. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arxiv*.

Hang Zhang, Xin Li, and Lidong Bing. 2023. [Video-LLaMA: An instruction-tuned audio-visual language model for video understanding](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 543–553, Singapore. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with bert](#). In *International Conference on Learning Representations*.

Yixiao Zhang, Junyan Jiang, Gus Xia, and Simon Dixon. 2022. [Interpreting Song Lyrics with an Audio-Informed Pre-trained Language Model](#). In *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, pages 19–26. ISMIR.

Zhi Zhong, Masato Hirano, Kazuki Shimada, Kazuya Tateishi, Shusuke Takahashi, and Yuki Mitsufuji. 2023. An attention-based approach to hierarchical multi-label music instrument classification. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

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

A Appendix

A.1 OpenMU-Bench data statistics

Detailed dataset statistics of OpenMU-Bench are shown in Table 5.

A.2 Training details and hyperparameters

In this section, we describe the detailed settings and hyperparameters used for training OpenMU. All experiments were conducted using 8-16 A100 40GB GPUs, with BF16 enabled to ensure stable training. We use DeepSpeed ZERO-3 (Rajbhandari et al., 2020) and Flash Attention 2 (Dao et al., 2022) to reduce the memory consumption. We utilized the Adam optimizer and a cosine learning rate scheduler, with a 30% warm-up ratio.

Figure 4 displays the training trajectories (log-scale) of both Stage (1) and Stage (2) training, where we apply mean-pooling to every 2–128 music tokens output by AudioMAE.

For Stage (1) training, we pretrained OpenMU for 15 epochs on the captioning subtask of OpenMU-Bench, which consists of approximately 275K pairs of music clips and corresponding captions. Stage (1) training took approximately 10 hours for the checkpoint we evaluated (i.e., mean-pooling every 8 music tokens as illustrated in Figure 4. The initial learning rate was set to 1e-3, with a batch size of 8 per GPU.

For Stage (2) training, we extended pretraining of OpenMU for 10 epochs on the captioning and reasoning subtasks of OpenMU-Bench, which comprise roughly one million training examples. The initial learning rate was set to 2e-5, with the same per-GPU batch size of 8. Stage (2) required approximately 40 hours due to the increased size of training data.

	Captioning		Reasoning		Lyrics		Tool-Use		MultipleChoice		Music Clips
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	
MusicCaps	2640	2839	-	-	-	-	-	-	-	-	5479
MusicInstruct	28670	30593	-	-	-	-	-	-	-	-	(5479)
LPMusicCaps	7920	-	-	-	-	-	-	-	-	-	(2839)
LPMusicMTT	51531	13386	-	-	-	-	-	-	-	-	25863
Music4all*	104268	5000	449711	21543	-	-	-	-	-	-	109269
MusicQA-Fin.	31116	-	38895	-	-	-	-	-	-	-	12543
MusicQA-Test	-	2240	-	2800	-	-	-	-	-	-	500
GTZAN*	639	290	2329	1116	-	-	-	-	-	-	1000
MusicNet*	3791	140	-	-	-	-	-	-	-	-	330
MTT*	-	-	78839	16100	-	-	-	-	-	-	(25863)
MTG-Jamendo*	45129	5144	177771	20308	-	-	-	-	-	-	50273
BART-Fusion	-	-	-	-	55262	800	-	-	-	-	14985
Tool-Using*	-	-	-	-	-	-	1612	403	-	-	0
MuChoMusic	-	-	-	-	-	-	-	-	-	1187	1187
Total	275704	54632	747545	61867	55262	800	1612	403	0	1187	221429

Table 5: OpenMU-Bench tasks and dataset distributions. “MusicQA-Fin.”: MusicQA-Finetuning. *: datasets with our new annotations. Numbers in brackets are not included when calculating the total number of music clips, as they represent captions annotated for the same set of music clips.

For the lyrics understanding subtask, we trained OpenMU for 20 epochs, reusing the hyperparameters from Stage (2). Similarly, for the tool using subtask, we reused the Stage (2) hyperparameters but reduced the number of epochs to 5 due to the smaller dataset size for this task.

A.3 More results

We show the BertScore performances of different OpenMU variants with various number of pooling music tokens and LoRA parameters in Figure 6. We also report the performance of OpenMU-1B in Table 6.

A.4 Metadata of datasets

In this paper, we contribute to creating the large-scale benchmark suite OpenMU-Bench for music understanding.

In contrast to other modalities such as images, where rich natural language descriptions are widely available across the internet (Schuhmann et al., 2022), music clips are often accompanied by tags, such as genre, year, and instruments. We consider these tags to be a form of metadata for the music clips. When constructing OpenMU-Bench, we bootstrap captions and reasoning texts in natural language about the music clips based on this metadata by prompting GPT-3.5.

Table 7 demonstrates the various types of metadata used in the OpenMU-Bench subtasks to create music understanding examples. Due to the broad coverage, music clips from different OpenMU-Bench subsets are associated with diverse types of metadata. Even within the same subtask, differ-

ent music clips may be tagged with only a limited set of metadata types. We detail how we process the metadata of each music clip as follows.

Tempo. Music clips in two datasets, music4all and GTZAN, are associated with tempo, and we convert the numerical values into natural language descriptions, following the Italian musical terms⁷ as shown in image Figure 7.

For **energy**, **valence**, **danceability**, which are float scores ranging from 0 to 1, we convert them into natural language descriptions using empirical thresholds of 0.3 and 0.7. Taking energy as an example, we consider a energy level s , where $s \geq 0.7$ as a high energy level, $0.7 > s \geq 0.3$ as a medium energy level, and $0.3 > s$ as a low energy level.

For **genre**, **mood**, and **instrument** of Music4all, MTT, MTG-Jamendo, we merge their original metadata by manual annotations and corrections for consistency. Concretely, we keep the top 50 tags of MTT and MTG-Jamendo, following the recommendation of the authors (Law et al., 2009a; Bogdanov et al., 2019), and use the top 166 tags of Music4all, as recommended by music4all_contrib⁸. The corrections involves actions such as de-compounding (“acousticguitar” → “acoustic guitar”), unifying (“Female vocalists” → “female vocal”), expanding (“synth” → “synthesizer”) the tags, and the resulting metadata tags are list as follows:

As a result, a JSON formatted metadata is created for each of the music clips:

⁷Italian musical terms: <https://www.musicca.com/musical-terms>.

⁸https://github.com/keunwoochoi/music4all_contrib

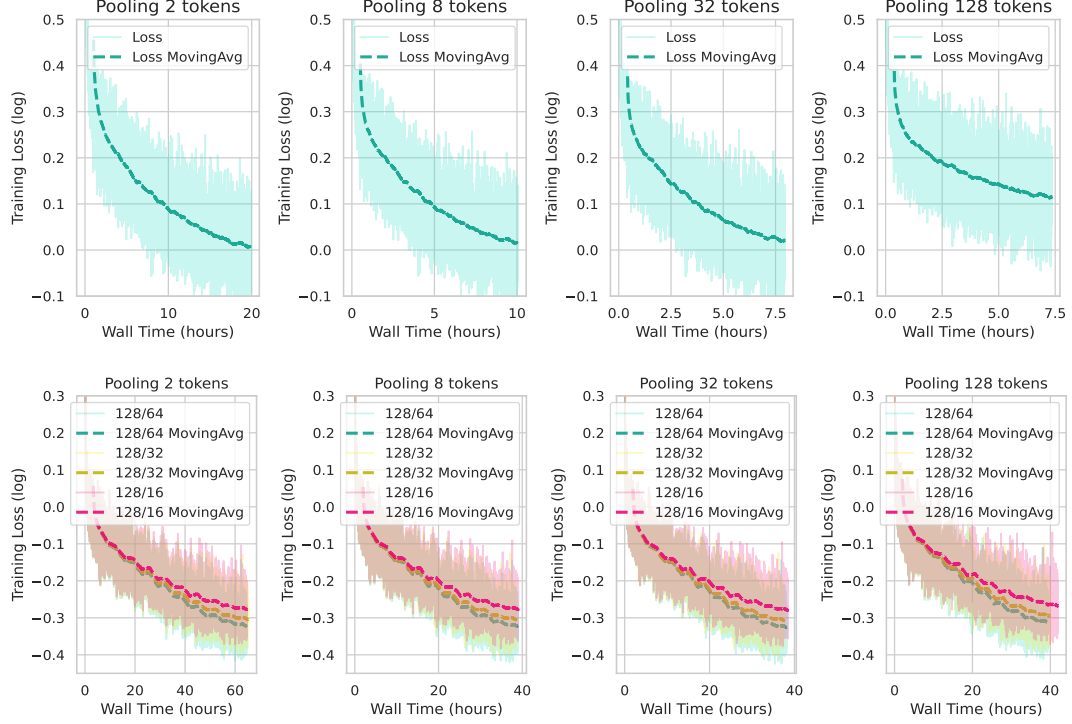


Figure 4: Training trajectories of Stage (1) (top) and Stage (2) (bottom). The x-axis represents the number of hours elapsed, and the y-axis shows the training loss on a log scale. We vary the number of mean-pooling music tokens from 2 to 128 and experiment with different LoRA parameter combinations, α/r . “MovingAvg” represents the moving average.

	BLEU-1				BLEU				RougeL				RougeL				BertScore				Meteor			
	OMU	OMU-1B	MUL	Qwen	OMU	OMU-1B	MUL	Qwen	OMU	OMU-1B	MUL	Qwen	OMU	OMU-1B	MUL	Qwen	OMU	OMU-1B	MUL	Qwen	OMU	OMU-1B	MUL	Qwen
MCaps	25.62	21.07	9.78	26.16	2.89	3.18	0.73	4.74	27.99	26.35	21.22	26.98	18.56	18.23	15.64	19.83	86.63	85.74	86.85	87.15	21.83	25.67	11.03	19.30
ML-S	18.92	15.35	50.01	35.70	8.75	7.25	23.90	16.00	41.03	32.01	52.69	46.97	37.75	30.19	47.73	41.85	89.90	88.13	92.83	91.82	43.81	40.98	49.10	45.14
ML-L	36.66	36.78	2.13	9.36	4.18	4.52	0.18	0.73	38.74	38.99	19.34	22.55	21.70	22.23	13.55	15.17	87.24	87.37	85.90	85.65	24.43	24.69	8.83	11.54
MTT	23.83	15.57	18.87	11.49	2.56	1.78	0.56	0.28	29.23	25.10	21.68	16.93	21.57	18.91	16.07	13.50	89.75	88.34	88.37	86.33	24.60	24.07	13.96	12.54
M4A*	51.03	51.10	5.11	12.06	18.69	18.84	0.36	0.58	51.31	51.41	19.51	17.55	34.80	34.91	14.31	12.80	91.04	90.80	86.64	83.96	43.58	43.37	10.07	10.27
MQA	19.60	21.76	19.65	15.64	2.22	4.37	5.08	1.40	23.96	26.13	31.74	18.72	17.39	19.63	28.07	15.15	87.30	87.66	89.45	85.57	26.34	28.16	21.73	13.53
GTZAN*	45.38	45.90	4.56	12.22	11.78	11.35	0.32	0.50	44.01	44.15	20.41	17.04	27.62	27.45	15.17	12.42	89.34	89.10	87.03	84.16	34.71	33.81	10.65	9.63
MNet*	52.68	51.60	1.17	4.83	22.14	20.91	0.00	0.18	56.44	55.46	14.49	12.60	38.11	37.20	12.25	10.71	91.97	91.76	85.59	83.94	45.92	44.69	6.56	6.34
MTG-J.*	47.56	48.82	5.32	12.10	15.83	16.38	0.41	0.67	49.66	50.13	20.29	16.30	33.76	34.22	15.94	12.64	90.79	90.73	87.72	85.20	39.44	39.24	10.06	9.53

	BLEU-1				BLEU				RougeL				RougeL				BertScore				Meteor			
	OMU	OMU-1B	MUL	Qwen	OMU	OMU-1B	MUL	Qwen	OMU	OMU-1B	MUL	Qwen	OMU	OMU-1B	MUL	Qwen	OMU	OMU-1B	MUL	Qwen	OMU	OMU-1B	MUL	Qwen
M4A*	49.20	49.31	18.13	21.3	23.31	23.39	5.97	6.53	53.26	53.29	34.11	31.66	41.08	41.17	25.01	23.32	92.34	91.52	89.63	88.25	49.96	49.37	22.51	31.31
MQA	24.84	29.08	40.64	27.6	9.46	11.59	22.47	12.87	35.86	38.58	51.29	45.20	30.66	33.33	47.54	40.93	89.70	88.82	92.59	90.93	40.04	40.5	46.15	43.35
GTZAN*	50.26	50.56	16.16	22.61	22.07	22.19	5.95	7.13	52.96	52.87	35.18	33.17	38.57	38.42	26.20	24.09	92.02	91.43	89.89	88.28	46.87	45.86	21.84	31.64
MTT*	45.52	46.91	21.70	25.18	21.18	21.90	8.31	8.30	50.83	51.32	38.70	33.35	39.93	40.26	29.92	25.64	92.03	91.43	90.63	88.93	48.06	47.61	26.33	29.40
MTG-J.*	45.87	47.00	23.69	21.82	21.12	21.64	8.59	6.96	50.78	51.13	38.45	32.19	39.74	39.92	29.27	24.57	92.01	91.14	90.47	88.52	47.97	47.60	26.42	30.70

Table 6: OpenMU-Bench captioning (top) and reasoning (bottom) results (in %) of OpenMU-8B (OMU), OpenMU-1B (OMU-1B), MU-LLaMA (MUL), and Qwen-Audio (Qwen). *: datasets with our new annotations.

```

{
  "dataset_name": "music4all",
  "audio_filename": "4MqXFtyr1XwxrShX.mp3",
  "tempo": "walking pace tempo",
  "valence": "medium valence",
  "energy": "high energy",
  "danceability": "medium danceable",
  "genre": [
    "rock",
    "pop",
    "electronic"
  ],
  "mood": [
    "ambient"
  ]
}

```

which is then employed to prompt GPT3.5 to create examples for music understanding, as described in the next section §A.5.

A.5 Prompts and dataset format

Based on the metadata of each music clip (§A.4), we prompt GPT-3.5 to generate examples for the music understanding tasks. Our prompts are adapted from those used by Gardner et al. (2024), with modifications tailored to the available metadata of different OpenMU-Bench subsets and sub-tasks. As an example, we present the prompt used for the music captioning task in Music4All:

	Tempo	Energy	Valence	Danceability	Genre	Mood	Instrument	Others
Music4all	○	○	○	○	△	△	△	△
GTZAN	○	×	×	×	○	×	×	×
MusicNet	×	×	×	×	×	×	○	×
MTT	×	×	×	×	△	△	△	△
MTG-Jamendo	×	×	×	×	△	△	△	△

Table 7: The metadata associated with each subtask dataset in OpenMU-Bench. ○: the metadata, e.g., tempo, is available for the music clips. △: the metadata *maybe* available for some music clips, but not for all of them. ×: the metadata is not available for the music clips.

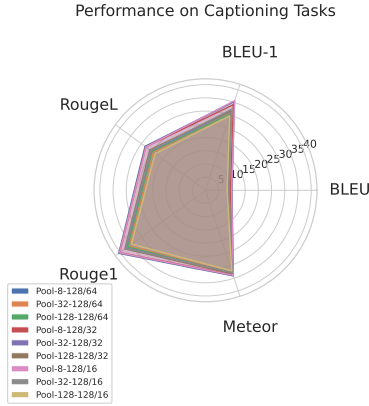


Figure 5: OpenMU variants performance on OpenMU-Bench captioning task. For each evaluation metric, such as BLEU, we report the macro average of the model’s performance across all OpenMU-Bench subtasks. “Pool-8-128/16” represents mean pooling every eight music tokens and using LoRA parameter $\frac{128}{16}$.

In short, we ask GPT-3.5 to generate an informative caption that can describe the key features and components of the music clip, based on its internal music knowledge accumulated during pretraining and our provided rich metadata. The generated results of the example in §A.4 is:

```
{
  "instruction": "Explain the contents of this song.",
  "output": "This music clip features high energy
    with a walking pace tempo. It combines elements
    from rock, pop, and electronic genres,
    creating a dynamic and lively soundscape.
    The medium valence contributes to a balanced
    emotional tone, while the medium danceability
    suggests a rhythmic and engaging quality,
    suitable for movement and dance. The ambient
    mood adds a layer of atmospheric texture to the
    overall vibrant and energetic composition.",
  "local_audio_path": "4MqXFtyr1XwxrShX.mp3",
  "task": "captioning",
  "dataset": "music4all_test"
},
```

which is then leveraged to train or test OpenMU according to the dataset split.

A.6 Dataset splits

In this section, we provide details about the train/test splits of the OpenMU-Bench subtasks.

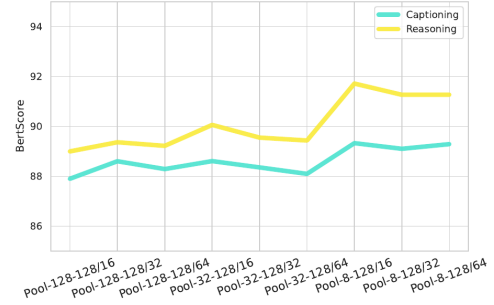


Figure 6: Performance of OpenMU variants on the captioning and reasoning tasks of OpenMU-Bench using BertScore as the metric.

Specifically, for MusicCaps, MusicInstruct, LP-MusicCaps, LPMusicMTT, MusicQA, MusicNet, BART-Fusion, and MuchoMusic, we follow the train/test splits proposed in the original papers. For GTZAN, we used the widely accepted filter-fault split (Kereliuk et al., 2015), and the split from MARBLE (Yuan et al., 2023) for MTT.

For Music4All, we start with the 800 music clips from BART-Fusion as the initial test set. We then expand this set by randomly sampling music clips until the total reaches 5,000. The remaining music clips and their annotations are used as training data. For MTG-Jamendo, we use annotations where the music clips from folds 90 to 99 of the original dataset (Bogdanov et al., 2019) serve as the test data, while the remaining clips and their annotations are treated as training data. For tool using, we randomly sample 80% examples for training and 20% for testing.

A.7 Tools

We define simple tools for solving MIR tasks such as tempo estimator. They are implemented as simple Python wrapper to the Madmom toolkit (Böck et al., 2016), which has been widely used in MIR. For example, the tempo estimator can be implemented as:

```
from madmom.features.beats import RNNBeatProcessor
from madmom.features.tempo import TempoEstimationProcessor
```

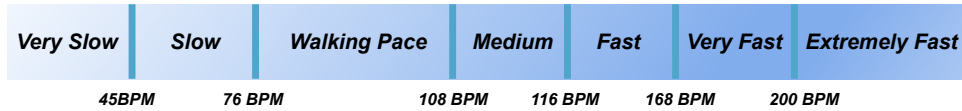


Figure 7: Converting numerical values (in beats per minute; BPM) of music tempo to natural language descriptions. The conversion is done based on the Italian musical terms.

Metadata of genre, instrument, mood, and others	
Genre:	instrumental, triphop, world, pop punk, hardcore, metalcore, mb, 70s, death metal, dream pop, brazilian, easylistening, classical, metal, rock, instrumental pop, 90s, dance, reggae, acoustic, 80s, orchestral, lounge, indie pop, british, electronic, pop, soul, experimental, hip-hop, indian, indie, indie rock, heavy metal, 60s, punk, progressive rock, synthpop, jazz, hard rock, post-hardcore, funk, alternative rock, new age, post-punk, pop rock, trance, mpb, pop folk, classic rock, techno, soundtrack, new wave, atmospheric, land, lo-fi, downtempo, rap, folk, opera, house
Instrument:	harpsichord, piano, strings, choral, flute, vocal, keyboard, violin, drums, computer, bass, harp, drum machine, acoustic guitar, electric guitar, no vocal, electric piano, synthesizer, cello, female vocal, guitar, male vocal, sitar
Mood:	psychedelic, soft, energetic, film, weird, ambient, loud, slow, chillout, relaxing, quiet, fast, happy, emotional
Others:	beats, solo, singer-songwriter

```

1236 def EstimateTempo():
1237     wav = load_audio(AUDIO_FILE)
1238     beat_proc = RNNBeatProcessor()
1239     tempo_proc = TempoEstimationProcessor(fps=100)
1240     beat_acts = beat_proc(wav)
1241     tempo_acts = tempo_proc(beat_acts)
1242     tempo_est = round(tempo_acts[0][0], 1)
1243     return tempo_est

```

OpenMU then calls for such a tool to estimate the tempo, when being asked questions such as “Let me know the tempo of this music clip.” and replying with “The music has tempo [EstimateTempo() → n] beats per minute.”.

Note that in our actual implementations, we use pseudo tool names, e.g., F1, for the tools. We found that the LLM tends to hallucinate new tools when camel case names like "EstimateTempo" are used, likely due to the presence of code in its pretraining data (Dubey et al., 2024).

Example prompt used for Music4all

You are an expert AI assistant that is knowledgeable about music production, musical structure, music history, and music styles, and you are hearing audio of a short clip or loop extracted from a piece of music. What you hear is described in a JSON-format shown below, describing the same audio clip you are listening to. This description is provided in a JSON dictionary, where the keys and values represent attributes of the music clip. The JSON contains a set of fields describing various features of the music clip.

Specifically, the JSON will contain:

- "tempo": the tempo of this music clip.
- "energy": the energy level of the music clip. High energy means fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on energy. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
- "valence": the valence level of the music clip. Valence measures the musical positiveness conveyed by the song. High valence value means the song is more positive (e.g. happy, cheerful, euphoric), otherwise the song is more negative (e.g. sad, depressed, angry).
- "danceability": danceable level of this music clip. Danceability measures how suitable a song is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity.

Optionally, the JSON may contain the following OPTIONAL annotations of the song:

- "genre": a list of genres of the music clip.
- "mood": a list of possible moods of the music clip.
- "instrument": a list of possible instruments used by the music clip.
- "others": other informative factors of the music clip.

You should only use the optional annotations when provided. Otherwise, you MUST NOT mention them in the description.

Based on the JSON, **provide a detailed musical description of the clip, from the perspective of a musical expert describing the clip as they hear it being played**. Use your knowledge about music to make sure to describe the musical style and contents, and any unique features of the music clip.

Describe the music clip using a tone of describing facts. DO NOT USE languages such as "I hear ...". Just provide musical captions to the clip.

Only provide details that are based on the provided metadata or your background knowledge of music as an intelligent AI assistant. Explain any musical concepts that would be unfamiliar to a non-musician. Do not specifically reference the provided metadata in the response; instead, respond as if you are hearing the song and reporting a rich description of what you hear. Keep in mind that the music clip is only a short clip, loop, or part of a song, and NOT THE COMPLETE SONG.

IMPORTANT!! DO NOT use the word "metadata" anywhere in your descriptions. DO NOT disclose that metadata about the song is provided to you. DO NOT use the phrase "song data" anywhere. Do not reveal that you know details of how the music clip was produced; instead, use phrases like "it sounds like XXX instrument" or "what I hear might be a YYY microphone". Now please provide the musical description of the clip, strictly following above requirements.