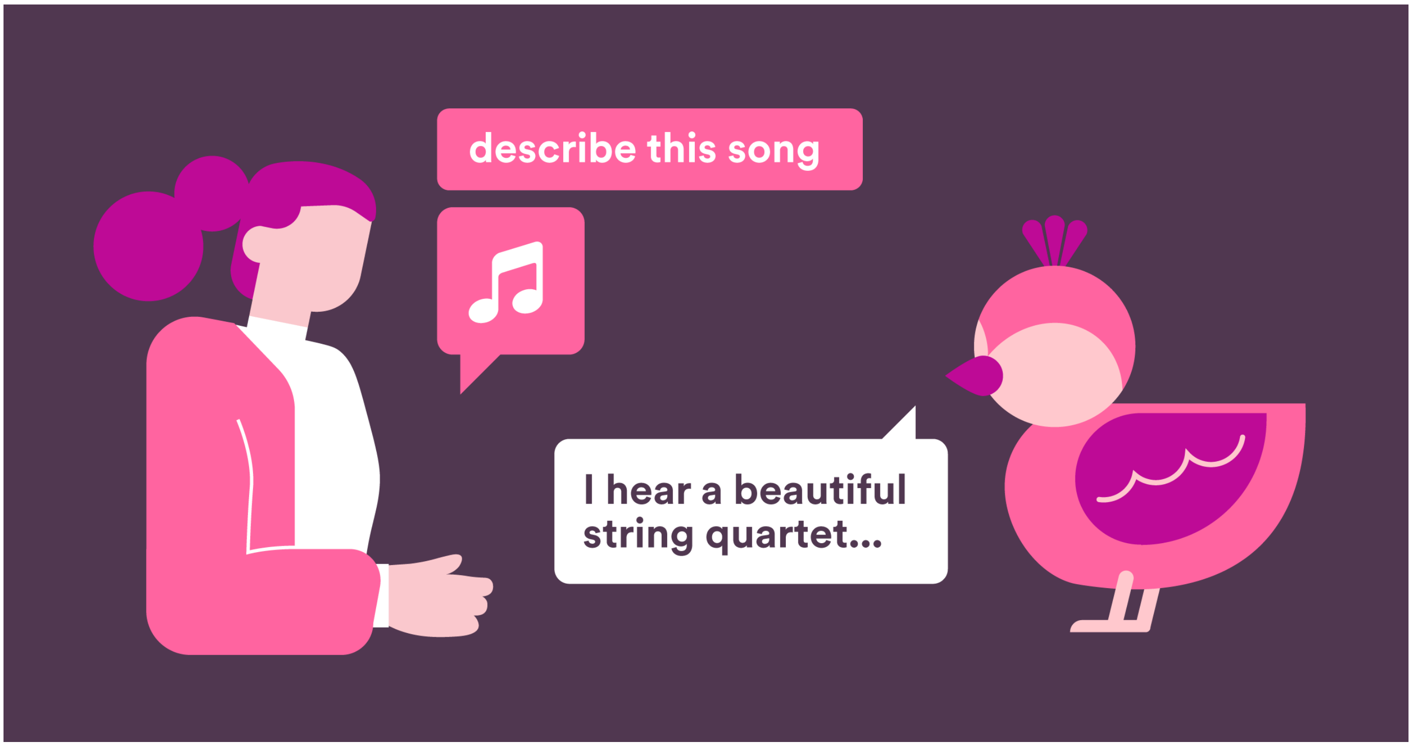
# LLark: A Multimodal Foundation Model for Music

October 13, 2023
Published by Josh Gardner, Simon Durand, Daniel Stoller, Rachel Bittner



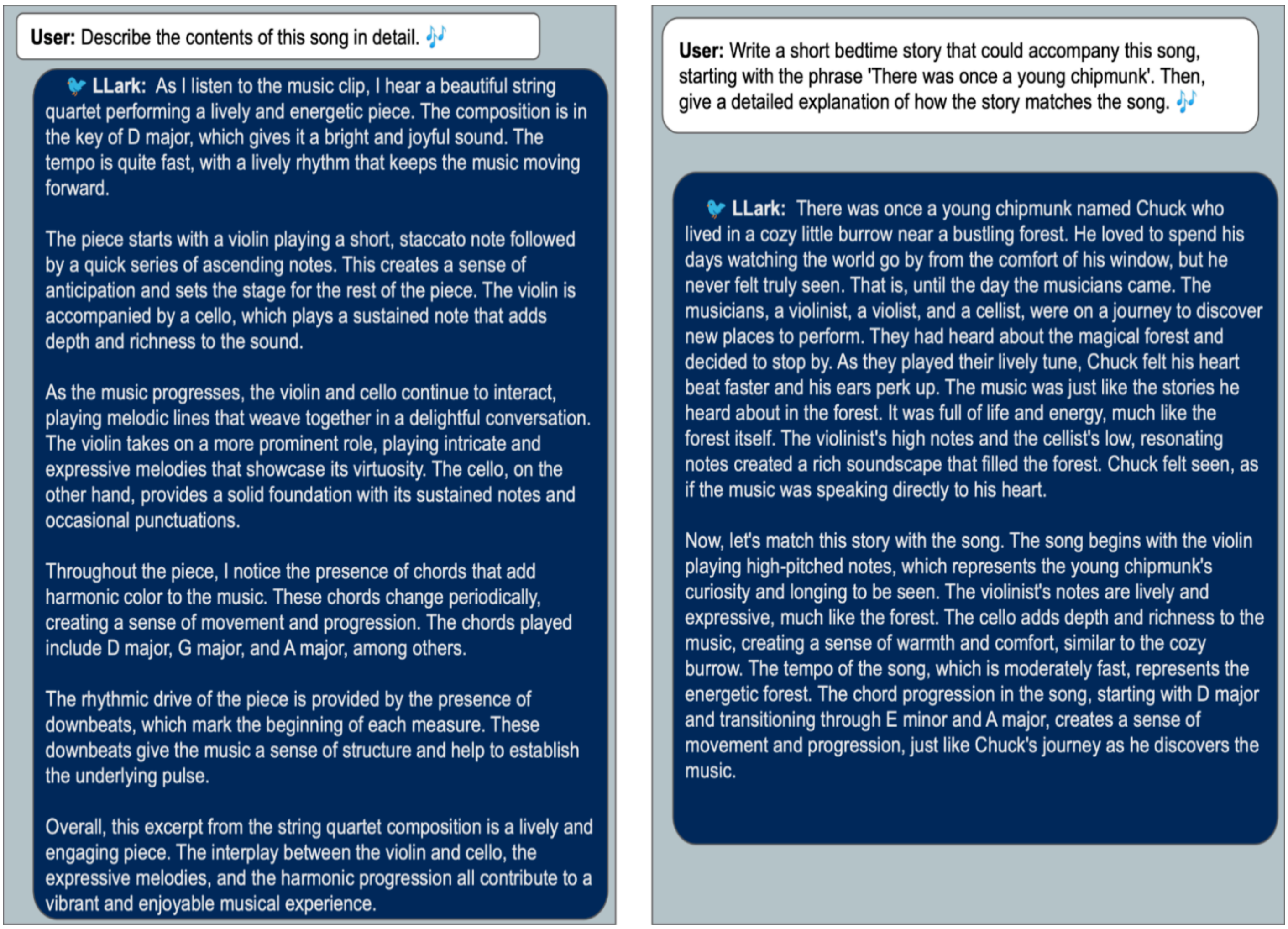

Every day, music is enjoyed, created, and discovered by billions of people around the globe – and yet, existing AI systems largely struggle to model the nuances that make music different from other forms of audio.

We are thrilled to introduce LLark, a multimodal language model for music understanding. In the world of music understanding research, there are many systems that can estimate specific individual attributes of a song (tempo, chords, etc.), and even generate short text descriptions of music (called "captions").

solve an incredible variety of tasks with a single system, just by simply asking a question or providing a set of instructions to a language model. Recently, this has expanded beyond pure text models, with AI systems able to dialogue about images, video, and audio. However, there are few models like this for music – and flexible multimodal language model for music understanding?

0:00 / 0:30

*Source: MusicNet/Museopen/Andrew D. Lawson*
*Title: Quartet No 53 in D major 3. Menuetto: Allegretto*
*License: Creative Commons Attribution 4.0 International*




*Example text inputs and outputs from LLark, for the provided audio. LLark is capable of both traditional music tasks, such as captioning (left), as well as more fun and complex tasks (right).*

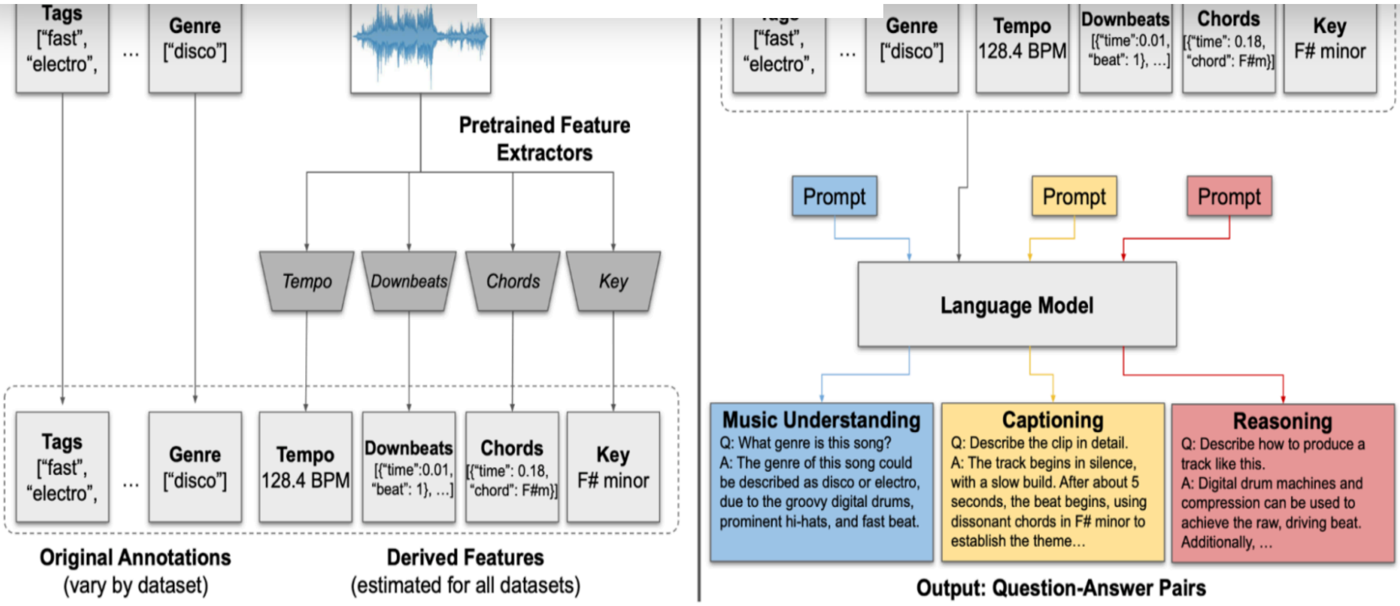
This blog post introduces the key ideas behind LLark, how we built it, and what LLark is capable of. If you are interested in finding out more details beyond this blog

## Building the Dataset

LLark is designed to produce a text response, given a 25-second music clip and a text query (a question or short instruction). In order to train LLark, we first needed to construct a dataset of (Music + Query + Response) triples.

We built our training dataset from a set of open-source academic music datasets (MusicCaps, YouTube8M-MusicTextClips, MusicNet, FMA, MTG-Jamendo, MagnaTagATune). We did this by using variants of ChatGPT to build query-response pairs from the following inputs: (1) the metadata available from a dataset, as pure JSON; (2) the outputs of existing single-task music understanding models; (3) a short prompt describing the fields in the metadata and the type of query-response pairs to generate. Training a model using this type of data is known as "instruction tuning." An instruction-tuning approach has the additional benefit of allowing us to use a diverse collection of open-source music datasets that contain different underlying metadata, since all datasets are eventually transformed into a common (Music + Query + Response) format. From our initial set of 164,000 unique tracks, this process resulted in approximately 1.2M query-response pairs.

We can illustrate our dataset construction process with an example. Given a song with the available tags "fast," "electro," and genre label "disco", we use off-the-shelf music understanding models to estimate the song's tempo, key, chords, and beat grid. We combine all of this metadata and ask a language model to generate one or more query-response pairs that match this metadata. For example, the model might generate the Query/Response pair: "How would you describe the tempo of this song?" → "This song has a fast tempo of 128.4 beats per minute (BPM)." We do this for three different types of queries (music understanding, music captioning, and reasoning) to ensure that LLark is exposed to different types of queries during training.

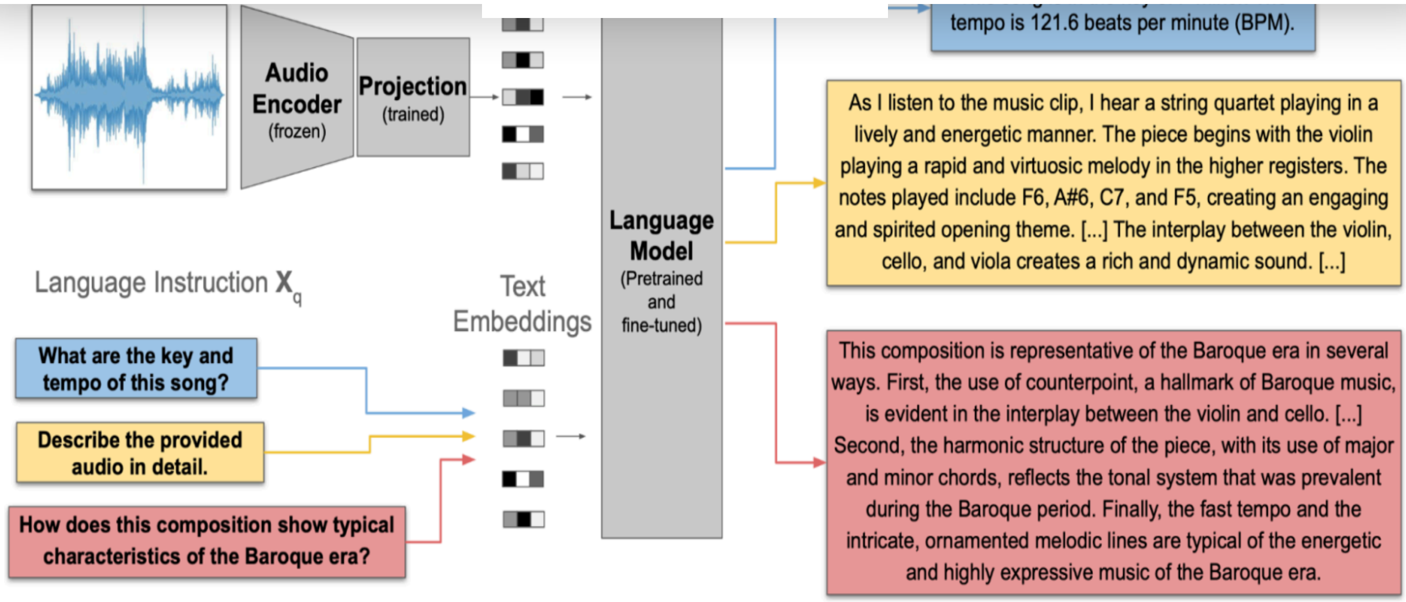



*An overview of LLark's dataset creation process.*

## Model Architecture and Training

LLark is trained to use raw audio and a text prompt (the query) as input, and produces a text response as output. LLark is initialized from a set of pretrained open-source modules that are either frozen or fine-tuned, plus only a small number of parameters (less than 1%!) that are trained from scratch.

The raw audio is passed through a frozen audio encoder, specifically the open-source Jukebox-5B model. The Jukebox outputs are downsampled to 25 frames per second (which reduces the size of the Jukebox embeddings by nearly 40x while preserving high-level timing information), and then passed through a projection layer that is trained from scratch to produce audio embeddings. The query text is passed through the tokenizer and embedding layer of the language model (LLama2-7B-chat) to produce text embeddings. The audio and text embeddings are then concatenated and passed through through the rest of the language model stack. We fine-tune the weights of the language model and projection layer using a standard training procedure for multimodal large language models (LLMs).

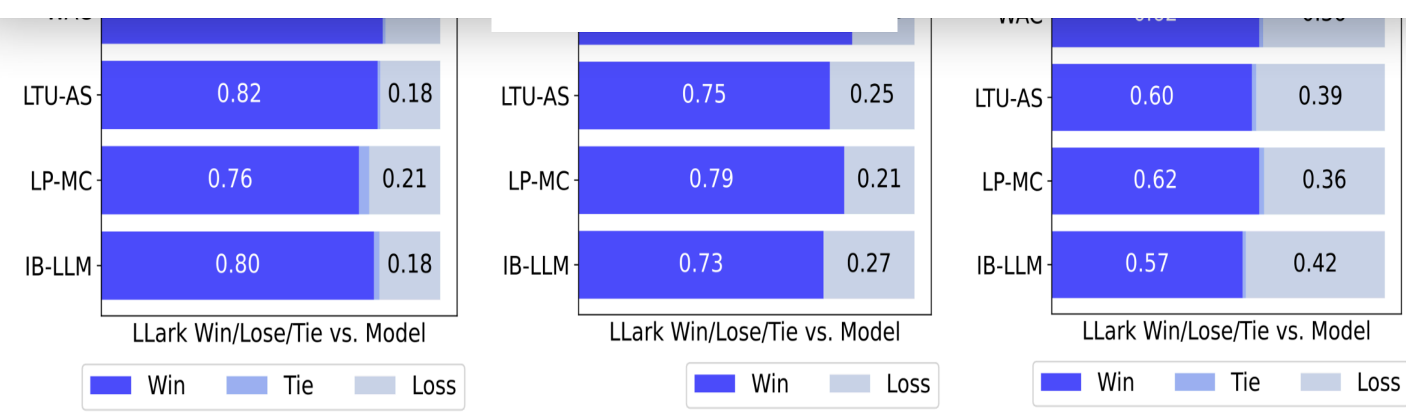



*An overview of LLark's model architecture. The blue, yellow and red boxes show example inputs and outputs in our three task families: music understanding, music captioning, and reasoning.*

## Evaluating LLark's Performance

We performed an extensive set of experiments to evaluate LLark's output compared to other open-source music and audio models.

In one set of experiments, we asked people to listen to a music recording and rate which of two (anonymized) captions was better. We did this across three different datasets with different styles of music, and for 4 different music captioning systems in addition to LLark. We found that people on average preferred LLark's captions to all four of the other music captioning systems.




*Win rate of LLark vs. existing music captioning systems.*

While the above experiments evaluate LLark's ability to describe music, they do not guarantee that the musical details LLark gives are accurate: it is possible that human raters cannot easily assess these details. We conducted an additional set of experiments to measure LLark's musical understanding capabilities. In these evaluations, LLark outperformed all baselines tested on evaluations of key, tempo, and instrument identification in zero-shot datasets (datasets not used for training). In zero-shot genre classification, LLark ranked second, but genre estimation is a difficult and subjective task; we show in the paper that LLark's predictions on this task tend to fall within genres that most musicians would still consider correct (e.g., labeling "metal" songs as "rock"). We provide a summary of the results below; see the paper preprint for details on the metrics and baselines used.

| Task | Dataset | Metric | Baseline | IB-LLM | LTU-AS | LLark |
|---|---|---|---|---|---|---|
| **Key Estimation** | GiantSteps-Key | MIREX Score | 0.32 | 0.05 | 0.00 | **0.70** |
| **Tempo Estimation** | GiantSteps-Tempo | ACC2 | 0.77 | 0.05 | 0.00 | **0.86** |
| **Genre Classification** | GTZAN | ACC@1 | 0.1 | **0.71** | 0.30 | 0.56 |
| | MedleyDB | | 0.125 | **0.57** | 0.38 | 0.56 |
| **Instrument ID** | MedleyDB | F1 | 0.25 | 0.25 | 0.24 | **0.31** |

prior multimodal system to date. These experiments also investigate how the different components of LLark – the audio encoder, language model, and training dataset size – affect its capabilities. If you're interested in the details of these experiments, check out the paper preprint. Our main conclusion is that each core aspect of Llark – the training data,  audio encoder, and language model – contributes critically to its overall quality.

Finally, we had a lot of fun exploring what LLark is capable of, and our paper only gives a glimpse of LLark's exciting capabilities. We include a few more examples of LLark's responses to queries, along with the audio it's answering about (all of these examples use the same song as input).

0:00 / 0:30

*Source: FMA Title: Summer Wind Artist: Cyclone 60 License: Attribution-Noncommercial-Share Alike 3.0 United States*


## Conclusions

This post introduced LLark, our new foundation model for music understanding. We believe Llark is a big step toward the next generation of multi-modal music research. However, it's important to note that LLark isn't perfect – it faces some of the same challenges we have seen across the AI community, including sometimes producing "hallucinations" (producing answers that may be vivid and overconfident but contain incorrect information).

their data available via Creative Commons, we are not able to release the training data or model weights. However, we do release our open-source training code to help the community to continue to advance this important and exciting research direction.

We hope that our work spurs further research into the development of AI models that can understand music. We also encourage the field to continue to develop high-quality open-source tools, datasets, and models so that members of the research community can build the next generation of tools and reliably measure their progress.

To learn more and find out details about our model and experiments, check out our paper preprint, code, and the associated website with more examples.

SHARE

CATEGORIES

Audio Intelligence

Language Technologies

Machine Learning

## Related articles

PODTILE: Facilitating Podcast Episode Browsing with Auto-generated Chapters

Personalizing Audiobooks and Podcasts with graph-based models

In-context Exploration-Exploitation for Reinforcement Learning

Newsroom Spotify Jobs Spotify.com

Spotify R&D Engineering Spotify R&D Design

Legal Privacy Cookies About Ads