EXPLORING GPT'S ABILITY AS A JUDGE IN MUSIC UNDERSTANDING

Kun Fang[1,2] **Ziyu Wang**[3,2] **Gus Xia**[2,3] **Ichiro Fujinaga**[1]

[1] Schulich School of Music, McGill University [2] Machine Learning Department, MBZUAI
[3] Computer Science Department, NYU Shanghai
kun.fang@mail.mcgill.ca, ziyu.wang@nyu.edu,
gus.xia@mbzuai.ac.ae, ichiro.fujinaga@mcgill.ca

ABSTRACT

Recent progress in text-based Large Language Models (LLMs) and their extended ability to process multi-modal sensory data have led us to explore their applicability in addressing music information retrieval (MIR) challenges. In this paper, we use a systematic *prompt engineering* approach for LLMs to solve MIR problems. We convert the music data to symbolic inputs and evaluate LLMs' ability in detecting annotation errors in three key MIR tasks: beat tracking, chord extraction, and key estimation. A *concept augmentation* method is proposed to evaluate LLMs' music reasoning consistency with the provided music concepts in the prompts. Our experiments tested the MIR capabilities of Generative Pre-trained Transformers (GPT). Results show that GPT has an error detection accuracy of 65.20%, 64.80%, and 59.72% in beat tracking, chord extraction, and key estimation tasks, respectively, all exceeding the random baseline. Moreover, we observe a positive correlation between GPT's error finding accuracy and the amount of concept information provided. The current findings based on symbolic music input provide a solid ground for future LLM-based MIR research. [1]

1. INTRODUCTION

Recent advancements in text-based Large Language Models (LLMs) have showcased their significant reasoning and knowledge retrieval capabilities across various domains, including music understanding. For instance, the standard GPT-4 model performs better than random on music theory questions [1]. This success raises the question of whether such text-based reasoning abilities could enhance Music Information Retrieval (MIR) tasks. From a psychological perspective, we are interested in how a *cognition module*, typically represented by a text-based LLM, can possibly interplay with a *perception module*, typically represented by an MIR network, to improve music understanding.

A key challenge in achieving this goal is the inherent difference between music and text modality, which typically requires aligning data in other modalities to text. Common strategies include either transforming all inputs into a unified modality [1, 2], or developing adapters tailored to other domains, such as MiniGPT-5 [3] and NextGPT [4]. Given the substantial data requirements and training costs involved in addressing cross-modality issues, we believe a practical initial step for LLM-based MIR research is to translate sensory inputs into symbolic representations and investigate the performance of text-based LLMs in a training-free way (e.g., prompt engineering [5]). This methodology allows us to assess how much cognition alone, without additional auditory perception, can enhance MIR tasks.

To this end, we propose a systematic prompt engineering method to assess the music understanding capabilities of text-based LLMs, focusing specifically on their ability to *detect errors* in MIR annotations. Each task input includes: 1) a music segment converted into MIDI or higher-level musical features, 2) a corresponding MIR annotation with deliberately inserted errors, and 3) a text prompt that introduces the MIR problem and outlines relevant musical concepts. The LLM's role is to pinpoint errors within the musical annotations, effectively acting as an MIR judge. In all the tasks, annotation errors are randomly applied at controlled rates, and prompts are crafted using common prompt engineering techniques. Additionally, we propose a *concept augmentation* strategy to evaluate the LLM's behavioral consistency in response to the musical concepts provided. This involves adjusting the occurrence of certain musical concepts in the prompt, such as replacing a musical term (e.g., pitch sequence) with a more general term (e.g., time series) to obscure a concept, or vice versa, to explore whether these changes influence the LLM's performance in predictable ways.

We carried out experiments using the GPT-3.5 model (hereafter, GPT), targeting three MIR tasks: beat tracking, chord extraction, and key estimation. The experiment results indicate that the error detection rates are higher than random, achieving scores of 65.20%, 64.80%, and 59.72%, respectively. Furthermore, the concept augmentation experiments show that GPT's performance broadly

[1] Code repository: https://github.com/kunfang98927/gpt-eval-mir

correlates with the amount of musical concepts introduced in the prompts. These findings suggest that GPT exhibits measurable music understanding capability, which sets a foundational baseline for future LLM-based MIR research. In sum, the contributions of the paper are as follows:

1. **We pioneer the integration of MIR problems with text-based LLMs.** Our approach utilizes prompt-engineering techniques for MIR error detection and adopts the symbolic music format to unify music and text modality, which does not require additional training.
2. **We perform a systematic study on GPT's abilities as a judge** in beat tracking, chord extraction, and key estimation tasks, demonstrating GPT's capability in solving MIR problems.
3. **We provide a solid ground for LLM-based MIR research.** The proposed methodology sets a baseline for future studies.

2. RELATED WORK

Recently, the advancements of text-based LLMs [6–8] have expanded beyond textual data, incorporating capabilities to interpret information from various other modalities. In the computer music domain, the research to combine text and audio LLMs is also popular. For example, ChatMusician is a text-based LLM, which focuses mainly on generating symbolic music in ABC notation [1]; MusicGen [9] and Coco-Mulla [10] are audio-based LLMs allowing text and symbolic music control; and MU-LLaMA is an audio-to-text model for caption generation [11]. Despite all these achievements, the current cross-modal research of text-based LLMs is restricted to generative tasks; and their ability to reason about cross-domain data is still under-researched. The focus of this paper is to evaluate whether LLMs can be used for music understanding and solving MIR problems.

In most cross-modal LLM studies, extensive training is required to align cross-modal information. These approaches involve training separate adapters to align the pre-trained model with other-domain data [3,4,12], fine-tuning an LLM on symbolic cross-domain data [1], or learning a trainable autoencoder to convert other-domain data to text tokens [2]. In the music domain, since music can be naturally represented as readable symbolic representations, we propose using prompt engineering methods to connect the music and text domains to avoid extra training.

The cross-domain prompt engineering methods used in this paper originate from the text domain. These strategies involve chain-of-thought [5], few-shot prompting [13], least-to-most prompting [14], and many others [15–17]. These methods show that the more organized the prompt is, the better the LLM will be able to reason. To the best of our knowledge, we present the first attempt of using prompt engineering to teach LLMs to reason about music. We aim to explore to what extent music reasoning alone can help MIR.

3. METHODOLOGY

In this study, we use prompt engineering to evaluate the capabilities of text-based LLMs through three MIR error detection tasks: beat tracking, chord extraction, and key estimation (as shown in Figure 1). In Section 3.1, we introduce the task definition and data representations for each task. In Section 3.2, we discuss the structure and main components of the prompts. Finally, Section 3.3 introduces the proposed concept augmentation methods to test the LLMs' music reasoning ability with respect to the music concepts included in prompts.

3.1 Task Definition and Data Representation

For symbolic MIR tasks, beat tracking determines the precise timing of beats in a MIDI-like music representation, chord extraction assigns a chord label to each segment, and key estimation identifies the musical key of each segment. Building on these tasks, we introduce a novel task: MIR error detection. This task involves identifying errors specific to each of the three traditional MIR tasks. The following subsections define the error detection tasks for beat tracking, chord extraction, and key estimation.

3.1.1 Beat Tracking Error Detection

We deliberately introduce a certain proportion of errors to the ground-truth beat annotations and ask the LLM to output the *beat index range* containing beat errors based on the music performance data in the symbolic music format. We introduce three types of error on beat annotations: 1) insert an extra beat between adjacent beats; 2) delete a beat; and 3) offset the timing of one beat, where the offset should be greater than a 70ms tolerance [18]. In beat tracking tasks, error detection is not a binary classification problem per detected beat, because there are false negative predictions (i.e., missed beats error). Therefore, it is crucial to return the beat index range so that both false positive beats and missed beats can be captured.

As shown in Figure 1 (left), the music segment and the beat annotations with errors are provided in the JSON format. The notes and beats are sorted by the temporal positions (i.e., onsets or beat locations).

3.1.2 Chord Extraction Error Detection

We deliberately introduce chord annotation errors and ask the LLM to output the *indices of incorrect chords* based on the music performance data in the symbolic music format. The chord errors are applied to the root, chord quality, and chord inversion attributes independently at a controlled rate.

As shown in Figure 1 (middle), we use the JSON format to represent the music segment and the chord annotations. The notes and chords are sorted by their temporal positions, and chords are notated as chord symbols in the conventional format [19].

3.1.3 Key Estimation Error Detection

We deliberately introduce key annotation errors and ask the LLM to output "correct" or "incorrect" based on the

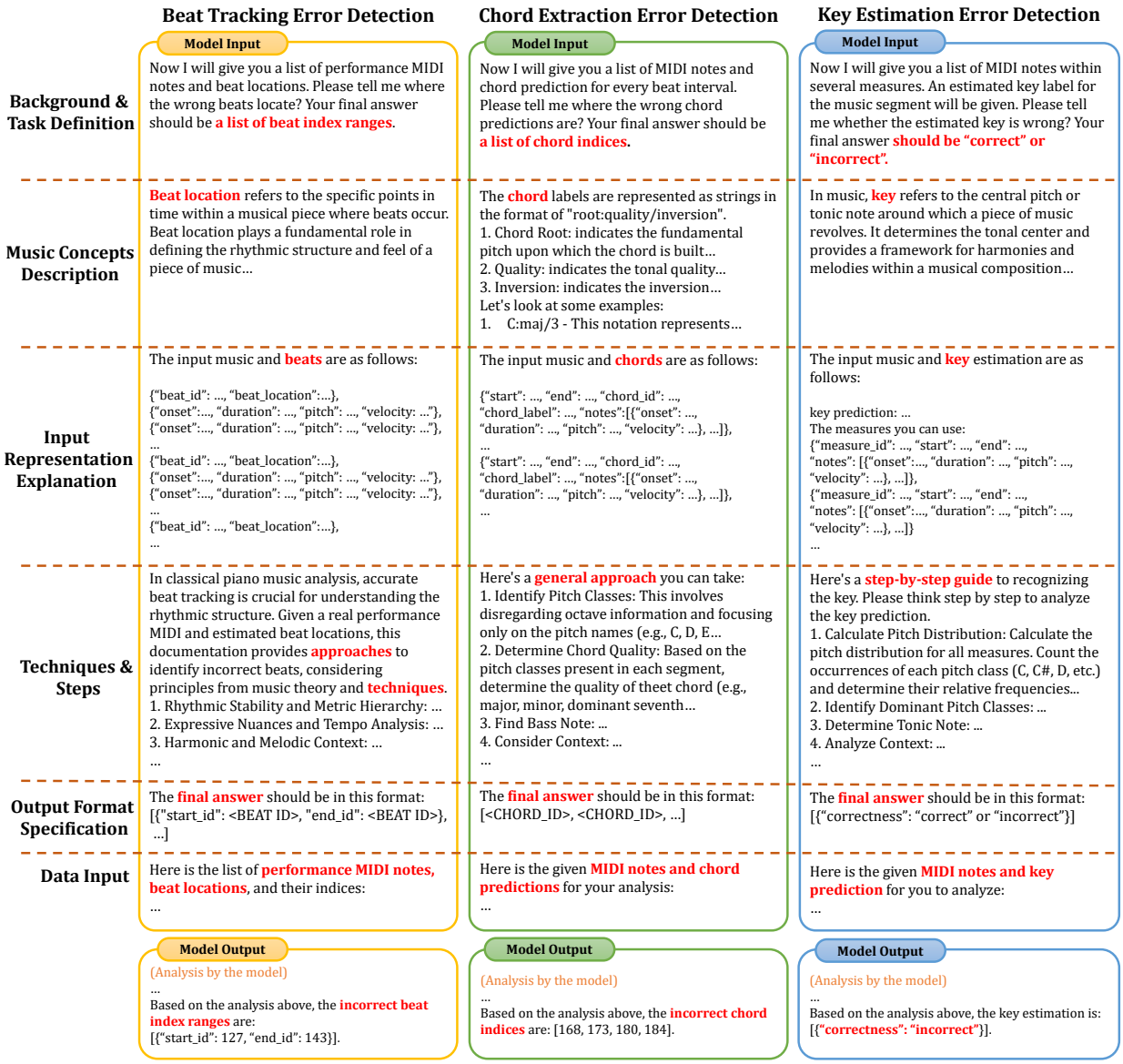

Figure 1: The example prompts and model outputs for the three error detection MIR tasks: beat tracking, chord extraction, and key estimation. Some keywords are highlighted in red in this figure for better readability. Orange texts indicate omitted content. The prompt structure is shown on the left.

music performance data in symbolic format. The errors are introduced by selecting an incorrect key out of the other 23 major and minor keys randomly at a controlled rate.

As shown in Figure 1 (right), we use the JSON format to represent the music segment and key annotations, where the key annotation is given at the beginning. The predicted key is represented by a formatted string of tonic and mode (e.g., "`A:min`").

3.2 Prompt Structure

Our investigation of prompt engineering methods indicates that a well-organized prompt structure is essential for successful MIR error detection. As shown in Figure 1, the prompt of the three MIR tasks all consists of six components as follows:

- **Background and Task Definition** introduces the MIR task and music domain background, and specifies the role of the LLM as a judge in assessing the correctness of MIR results.
- **Music Concepts Description** introduces relevant music concepts about beat, chord, or key, together with examples of those concepts. For example, we show examples of chord root, quality, and inversion for chord extraction, to guide the LLM to better parse the chord labels such as `C:maj/3`.
- **Input Representation Explanation** specifies the data structure and format of the input music data.
- **Techniques and Steps** provides reference steps to encourage the LLM to apply "chain-of-thought" in the error detection process. For example, we provide clear steps in the chord extraction task: 1) Identify Pitch Classes; 2) Determine Chord Quality; 3) Find Bass Note; 4) Consider Music Context; and etc.
- **Output Format Specification** defines the JSON-like output format, ensuring consistency for post-processing.

- **Data Input** provides the subject music piece and the MIR results to be judged, following the format defined in input representation explanation section.

3.3 Concept Augmentation

The prompts defined in Section 3.2 contain extensive music concepts for each of the three tasks, which we regard as *Basic Concepts*. Based on these, we apply *concept augmentation* by either introducing new concepts or masking basic concepts in order to compare the LLM performance under varying amounts of music knowledge provided.

In *Concept Introduction*, we add new concepts that are supposedly helpful for doing MIR tasks. For example, for beat tracking, we introduce "rhythm" to the LLM: we provide a brief description of on-beat notes and off-beat notes, and how to compute their density percentages. We explain how such concepts contribute to better judgments.

Conversely, we also define the *Concept Masking* operation, which eliminates or blurs music concepts at different levels. Such operations are used to examine the *innate* reasoning ability of LLMs as a reference:

- **Music Attribute Masking**: removes explanations about music concepts pertaining to the musical objects under operations. For example, in chord extraction, "root note", "chord quality", and "inversion" are replaced by an abstract generic expression, "a chord feature".
- **Task Masking**: on top of Music Attribute Masking, removes explanations about all concepts related to the MIR task, so that the LLM is required to reason about the correctness for a novel abstract task. For example, for beat tracking, the task becomes "Please read a sequence of MIDI notes and music labels to determine the correctness of each label." All expressions that imply beat tracking will be deleted, including "tempo", "fast", "slow", etc., to ensure that the task information is not implied in any form.
- **Domain Masking**: on top of Task Masking, eliminates explanations about all concepts related to the *music* domain to the greatest extent, leaving the LLM with an abstract logic-domain reasoning problem. For example, the LLM is told: "You will be given some labels and the corresponding raw data. Your task is to tell me where the wrong labels are located?"

4. EXPERIMENTS

We conduct our LLM-based MIR tasks with GPT-3.5. We introduce the datasets in Section 4.1 and the evaluation metrics in Section 4.2. The evaluation results are provided in Section 4.3.

4.1 Datasets

We use symbolic performance MIDI dataset for the three proposed tasks. For beat tracking error detection, we use classical piano recordings from the MAPS database [20], specifically from the "ENSTDkCl" subset (29 pieces) which has been commonly used as a beat tracking test set. We also use the corresponding metrical annotations from [21]. We randomly create beat errors by inserting (9%), deleting (12%), or offsetting (9%) beats, resulting in an emulated MIR prediction with an F-score of 0.8370.

For both chord extraction and key estimation error detection, we use MIDI for Chinese pop songs on a subset of the POP909 dataset [22]. For chord extraction, we randomly introduce errors in root, quality, or inversion with a ratio of 30%, resulting in an "MIR" accuracy of 0.7327. We choose 50 songs and divide each song into segments with 32 chord labels. For key estimation, we test on 757 songs in the dataset whose ground-truth key is unchanged throughout the piece. We randomly select three four-measure segments for each song and modify 30% of the key labels at random. A summary of the data statistics is shown in Table 1.

	Beat Tracking	Chord Extraction	Key Estimation
#Notes	70,607	48,919	177,535
#Labels	14,194	9,200	2,271
#Tokens (per call)	6065.31	9256.80	3214.63

Table 1: Statistics of the music data used for evaluation. The row #Notes represents the total number of MIDI notes processed for each task. The row #Labels indicates the number of labels used in the evaluation of each task. The row #Tokens (per call) shows the average number of tokens per call fed into the GPT-3.5 model for each task.

4.2 Evaluation Metrics

We design metrics to evaluate the performance of LLMs in identifying errors in MIR annotations. Since our approach does not directly predict MIR annotations, our metrics differ from existing MIR evaluation metrics. For chord extraction and key estimation tasks, we regard error detection as a binary classification task in which each chord or key label is classified as correct or incorrect. We use weighted precision, recall, and F1-score to evaluate GPT's performance on both correct and incorrect classes [23].

In beat tracking error detection, the beat sequence with potential errors is typically not one-to-one aligned with the ground truth beats. We consider three types of beat locations: 1) correctly identified beats, 2) additional beats, and 3) missing beats, which are also referred to as true positives, false positives, and false negatives, respectively, in conventional beat tracking tasks [18]. We use TP, FP, and FN to denote these sets of beat positions and I to denote the union of time intervals predicted by an LLM error detector. We consider the following metrics:

- **CPR (Correct Pass Rate on TP)** is defined as $\frac{|\text{TP}-I|}{|\text{TP}|}$, which measures the proportion of true positives that are correctly identified (by GPT) as "correct beats".
- $\textbf{EDR}_P$ **(Error Detection Rate on FP)** is defined as $\frac{|I\cap\text{FP}|}{|\text{FP}|}$, which evaluates the proportion of false positives that are correctly identified (by GPT) as "incorrect beats".
- $\textbf{EDR}_N$ **(Error Detection Rate on FN)** is defined as: $\frac{|I\cap\text{FN}|}{|\text{FN}|}$, which evaluates the proportion of false negatives that are correctly identified (by GPT) as "incorrect beats".

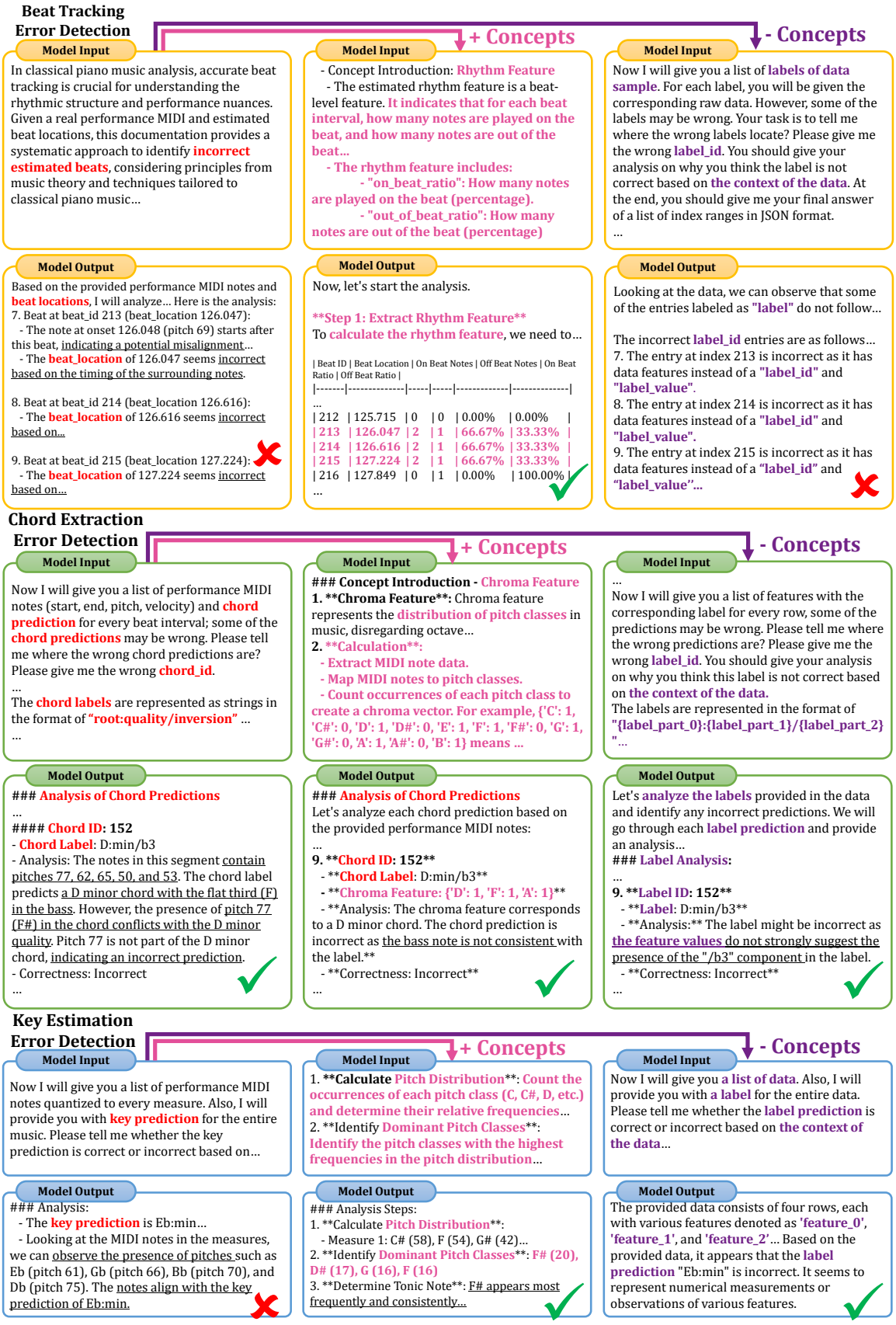

Figure 2: The impact of concept augmentation on GPT's behavior in three MIR error detection tasks: 1) *Basic Concepts* (left), 2) *Concept Introduction* (middle), and 3) *Concept Masking*: all music domain concepts removed (right). Red color indicates the basic concepts. Pink color indicates the introduced concepts. Purple color represents the expression after masking all music-related concepts. Underlines denote reasoning process. The checkmark indicates a correct judgment made by GPT, while the cross indicates an incorrect judgment by GPT.

Concept Augmentation	CPR↑	EDR_P↑	EDR_N↑	WS↑
Basic Concepts	0.6681	0.3728	0.1794	0.5607
+ "Rhythm"	**0.8533**	0.1496	0.0968	**0.6520**
- "Beat Location"(Music Attribute Masking)	0.6898	0.3720	0.2008	0.5792
- "Beat Tracking"(Task Masking)	0.5998	0.4010	0.2862	0.5296
- "Music"(Domain Masking)	0.2418	**0.7657**	**0.7061**	0.3785
Random	0.513 ± 0.0586	0.4891 ± 0.0564	0.3238 ± 0.0608	0.4843 ± 0.0274

(a) Evaluation results on beat tracking error detection

Concept Augmentation	$p\uparrow$	$r\uparrow$	$f\uparrow$
Basic Concepts	0.6345	0.6948	0.6207
+ "Chroma"	**0.6996**	**0.7174**	0.6290
- "Root"; "Quality"; "Inversion"(Music Attribute Masking)	0.6503	0.6992	0.6376
- "Chord Extraction"(Task Masking)	0.6497	0.6947	0.6362
- "Music"(Domain Masking)	0.6848	0.7144	**0.6480**
Random	0.5812 ± 0.0032	0.5003 ± 0.0034	0.5213 ± 0.0033

(b) Evaluation results on chord extraction error detection

Concept Augmentation	$p\uparrow$	$r\uparrow$	$f\uparrow$
Basic Concepts	0.5789	**0.6513**	0.5965
+ "Scale"	0.5847	0.6169	**0.5972**
- "Tonic"; "Mode"(Music Attribute Masking)	0.5754	0.5812	0.5782
- "Key Estimation"(Task Masking)	0.5840	0.6143	0.5960
- "Music"(Domain Masking)	**0.5927**	0.4161	0.4085
Random	0.5779 ± 0.0086	0.4977 ± 0.0093	0.5186 ± 0.0089

(c) Evaluation results on key estimation error detection

Table 2: The evaluation results of GPT on three MIR error detection tasks: beat tracking, chord extraction, and key estimation. Each task is assessed under different concept augmentation. "+" denotes *Concept Introduction*. "–" denotes *Concept Masking*. ↑ indicates that higher values are better. p, r, and f stand for precision, recall, and F-score, respectively.

Finally, we compute a weighted average of these metrics, denoted by WS:

$$\mathrm{WS} = \frac{\mathrm{CPR} \times |\mathrm{TP}| + \mathrm{EDR_P} \times |\mathrm{FP}| + \mathrm{EDR_N} \times |\mathrm{FN}|}{|\mathrm{TP}| + |\mathrm{FP}| + |\mathrm{FN}|}. \tag{1}$$

4.3 Evaluation Results

We evaluate the performance of GPT on three MIR error detection tasks. We first use the prompt with Basic Concepts and compare it with a random baseline, as well as prompts under different concept augmentation methods (see Section 3.3). The results are summarized in Table 2.

The results of beat tracking error detection task are shown in Table 2a. The random baseline is implemented by first randomly selecting k beat labels and joining consecutively selected beats into time intervals serving as detected error ranges. In Concept Introduction, we guide the GPT to compute the number of on-beat and off-beat note percentages, and in Concept Masking, we apply music attribute, task, and domain masking incrementally. Results show the basic prompt outperforms the random baseline in all prompt settings. Moreover, as the number of concepts decreases, the performance of GPT in judging the correctness of beat labels shows an overall downward trend.

The results of chord extraction error detection task are shown in Table 2b. The random baseline detects incorrectness with a probability of 50%. In Concept Introduction, we show GPT the chord chroma concept and encourage GPT to deduce the pitch distribution from input music. Results show that all GPT settings far exceed the random baseline. There remains a downward trend as the number concepts decreases except in the Domain Masking setting.

The results of key estimation error detection task are shown in Table 2c. The random baseline and concept augmentation are implemented similarly to those of chord extraction. In Concept Introduction, we show GPT the scale concept. Results show that GPT performs slightly better than the random baseline in F-score and recall, and similar to the baseline in precision. The downward trend of concept augmentation is less salient.

Finally, we provide a case study (Figure 2) to illustrate GPT's behavior under different settings of concept augmentation. In all tasks, GPT exhibits general time series analysis abilities even when music concepts are all masked, and the introduced music concepts help GPT to reason in a more musical fashion, particularly in beat tracking. However, we also observe limitations, including high randomness in output, sensitivity to prompts, and hallucination [24]. These issues make it challenging to empirically summarize or conjecture GPT's reasoning abilities in solving MIR problems in general.

5. CONCLUSION AND FUTURE WORK

In conclusion, we have proposed a methodology to solve MIR problems with text-based LLMs with prompt engineering. We evaluate the performance of GPT-3.5 in error detection across three MIR tasks and find out that GPT's music reasoning ability in MIR tasks can be enhanced when provided with well-structured prompts with music concepts. Across all three MIR error detection tasks, GPT consistently outperforms random baseline methods and demonstrates improved performance when prompted with additional music knowledge. In this study, we establish a baseline for assessing LLMs' ability to understand music solely through reasoning, paving the way for future LLM-based MIR research. In the future, we will consider evaluating LLMs' judging ability on real MIR errors instead of synthetic ones and using fine-tuning techniques to better explore LLM-based MIR study.

6. ACKNOWLEDGMENTS

This research has been supported by the Social Sciences and Humanities Research Council of Canada (SSHRC 895-2022-1004) and the China Scholarship Council.

7. REFERENCES

[1] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou *et al.*, "Chatmusician: Understanding and generating music intrinsically with llm," *arXiv preprint arXiv:2402.16153*, 2024.

[2] L. Yu, Y. Cheng, Z. Wang, V. Kumar, W. Macherey, Y. Huang, D. Ross, I. Essa, Y. Bisk, M.-H. Yang *et al.*, "Spae: Semantic pyramid autoencoder for multimodal generation with frozen llms," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[3] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.

[4] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua, "Next-gpt: Any-to-any multimodal llm," *arXiv preprint arXiv:2309.05519*, 2023.

[5] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[6] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[7] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, vol. 2, no. 3, p. 6, 2023.

[8] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[9] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023.

[10] L. Lin, G. Xia, J. Jiang, and Y. Zhang, "Content-based controls for music large language modeling," *arXiv preprint arXiv:2310.17162*, 2023.

[11] S. Liu, A. S. Hussain, C. Sun, and Y. Shan, "Music understanding llama: Advancing text-to-music generation with question answering and captioning," *CoRR*, vol. abs/2308.11276, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2308.11276

[12] H. Zhang, X. Li, and L. Bing, "Video-llama: An instruction-tuned audio-visual language model for video understanding," *arXiv preprint arXiv:2306.02858*, 2023.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[14] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le *et al.*, "Least-to-most prompting enables complex reasoning in large language models," *arXiv preprint arXiv:2205.10625*, 2022.

[15] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2022.

[16] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba, "Large language models are human-level prompt engineers," *arXiv preprint arXiv:2211.01910*, 2022.

[17] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022.

[18] M. E. Davies, N. Degara, and M. D. Plumbley, "Evaluation methods for musical audio beat tracking algorithms," *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009.

[19] C. Harte, M. B. Sandler, S. A. Abdallah, and E. Gómez, "Symbolic representation of musical chords: A proposed syntax for text annotations." in *ISMIR*, vol. 5, 2005, pp. 66–71.

[20] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2009.

[21] A. Ycart, E. Benetos *et al.*, "A-maps: Augmented maps dataset with rhythm and key annotations," 2018.

[22] Z. Wang, K. Chen, J. Jiang, Y. Zhang, M. Xu, S. Dai, X. Gu, and G. Xia, "Pop909: A pop-song dataset for music arrangement generation," *arXiv preprint arXiv:2008.07142*, 2020.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[24] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, mar 2023. [Online]. Available: https://doi.org/10.1145/3571730