A local LLM-based system for interactive musical improvisation

Olivier Jambois *
Department of Music and Sound
Escola de Noves Tecnologies Interactives - University of Barcelona, Spain
olivier.jambois@enti.cat

Abstract

We present a system based on a local Large Language Model (LLM) for interactive musical improvisation, addressing the trade-offs between reasoning, latency, and sound rendering. While cloud-based LLMs offer high reasoning capabilities, to bypass their unpredictable latency, we developed a local pipeline using a Llama-3.1-8B model as a drum pattern generator and contextual arbitrator for tempo and micro-timing parameters. The system employs a parallel dual-window analysis for audio descriptor extraction and tempo tracking via a Gaussian Mixture Model. The system prompt and the dynamic user prompt—which is periodically fed with the performance's audio descriptors—have been heuristically optimized to ensure task consistency and mitigate hallucinations. The LLM's symbolic output is routed to a Pure Data sequencer for rhythmic placement, then to a RAVE neural synthesizer for real-time timbral re-synthesis. The system's interactive potential was evaluated through performer studies with professional musicians. The main contribution of this work is the development of a portable, hybrid architecture that leverages the probabilistic reasoning of a local 8B LLM and the timbral richness of neural synthesis to facilitate a non-deterministic, organic dialogue between human and AI in the context of free improvisation.

1 INTRODUCTION

The integration of LLMs into real-time performance opens new avenues for musical co-creation, building on significant advances in deep generative models (Huang et al., 2019; Agostinelli et al., 2023; Ji et al., 2023). While the field of autonomous agents for improvisation has a long history (Assayag et al., 2006; Tatar and Pasquier, 2019), the deployment of these modern generative models in a live context introduces specific technical constraints. In the realm of free improvisation, where the dialogue between performers is guided by intuition, gesture, and shared idioms (Bailey, 1993), a significant challenge lies in leveraging these "black-box" reasoning engines into responsive, low-latency agents capable of maintaining a coherent musical narrative within the immediacy of a live performance.

In previous research, we explored the potential of using cloud-based Large Language Models (LLMs), specifically GPT-4o-mini, to leverage their cognitive and reasoning capabilities within a musical improvisation context (Jambois et al., 2026). The system aimed to determine if an LLM could act as a responsive musical partner. In this setup, the LLM was assigned the role of a percussionist, managing a three-instrument drum kit. We prioritized rhythmic generation over harmony, as rhythm provides a more direct entry point for evaluating an LLM's capacity for structural (adhering to musical composition idioms) and symbolic interaction.

Despite promising initial results, several critical limitations emerged. First, the reliance on standard beat-tracking (Librosa) within short analysis windows led to erratic tempo fluctuations, resulting in a non-musical "random walk" effect. Furthermore, using cloud-based models introduced unpredictable latency, varying from 2 seconds to over 10 seconds depending on network conditions, which is prohibitive for live performance. Finally, the use of static drum samples limited the system's expressive range, creating a mechanical sound that lacked the organic nuance required for musical expression.

To address these challenges, this paper presents a revamped, autonomous system designed for local execution on a laptop without the need for internet connectivity. The three main improvements are as follows. First, we introduce a dual-window analysis strategy utilizing a Gaussian Mixture Model (GMM) combined with the Bayesian Information Criterion (BIC) to stabilize rhythmic detection. Next, we benchmark and integrate a local Llama-3.1-8B model, identifying an optimal trade-off between reasoning, stability, portability, and a consistent 2-second inference latency when the prompt is optimized. Ultimately, we replace static samples with RAVE (Realtime Audio Variational auto-Encoder) neural synthesis (Caillon and Esling, 2021), providing the LLM with a more timbrally rich voice.

The resulting architecture is a hybrid system combining symbolic representation, LLM-driven contextual decision-making, and neural audio synthesis. This study demonstrates that this architecture based on an 8B-parameter model can effectively navigate the trade-off between limited computational resources and high-quality generative output, by offloading formal arithmetic tasks. Of particular significance, this allows the model to prioritize the musical intentionality, fostering a non-deterministic interaction where the system functions as an adaptive, unpredictable partner rather than a rigid accompaniment tool. After outlining the methodology, the results and limitations are presented and discussed.[1] Lastly, perspectives and future work are proposed.

*webpage: olivierjambois.com

[1]All supplementary audio examples and the full video referenced in this paper are archived on https://doi.org/10.5281/zenodo.21805358. The video is also available on https://youtu.be/AbmzL6wLgBQ for instant viewing.

2 METHODOLOGY

2.1 System pipeline

Figure 1 illustrates the proposed system architecture. It operates as a real-time interaction loop where the guitar input is captured via a Pure Data patch. Audio analysis is performed using two distinct temporal scales: a shorter 3s window and a longer 11s window, each optimized for specific audio descriptors. The resulting symbolic representation of the performance is transmitted to a Large Language Model (LLM), which generates a drum pattern, a tempo value, a "human feel" (micro-timing) parameter and the overall drums volume for the generated pattern. These values are then sent back to a second Pure Data patch, which hosts a drum sequencer coupled with a RAVE neural synthesizer. The resulting drum sounds are then output through a speaker, closing the interaction loop between the system and the musician.

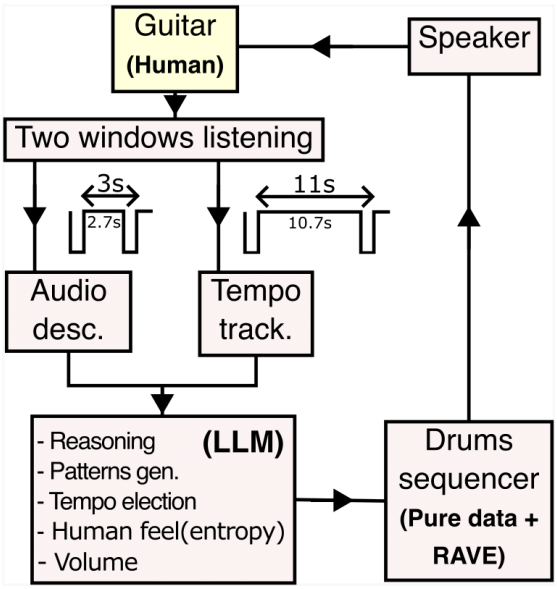

Figure 1: System architecture for human-AI musical co-creation. The pipeline illustrates the interaction loop between a human guitarist and an LLM-based drum sequencer.

2.2 Local LLM integration

Interaction with the LLM is based on a symbolic representation of the guitar performance, which is significantly more computationally efficient than processing raw audio signals. To enhance system portability and reliability, we opted for a local LLM rather than cloud-based solutions like GPT-4. This approach eliminates dependencies on internet connectivity and mitigates the risk of unpredictable network latency. The proposed system was evaluated on a workstation equipped with an NVIDIA RTX 4060 with 8GB of VRAM. To balance portability, performance and latency, we implemented a local Llama-3.1-8B model (4-bit quantization, inference time about 1-2 seconds). Comparative tests justifying this choice are presented in Section 3.1.

To maintain musical coherence, exchanges between the user and the LLM are cached in a local file and added to the conversation history (including both audio descriptors and generated patterns). However, retaining the full session history and sending it at each new prompt is infeasible, as the accumulating token count would increase computational load and latency. To manage this, we implemented a sliding window buffer that includes only a fixed number of recent exchanges. Preliminary observations indicated that while a three-exchange history did not degrade significantly latency, it occasionally caused the model to deviate from the constraints defined in the system prompt, specifically increasing the rate of hallucinations. Consequently, a two-exchange history was selected. This strategy helps to maintain a stable "narrative" continuity without risking computational overload as the session progresses.

2.3 Audio synthesis and output

The generated drum patterns consists of three instruments, kick, snare, and hi-hat, representing the core elements of a standard drum kit. The LLM generates a one-bar 4/4 pattern with sixteenth-note subdivisions for each instrument, resulting in three sequences of 16 binary values (0 for silence, 1 for onset). While we experimented with generating longer structures (e.g. two-bar patterns) to allow the LLM to compute the subsequent sequence while Pure Data reads the actual one, this approach was discarded, as the increased sequence length significantly impacted the latency and the hallucination rate. To move beyond the mechanical sound of static samples, the sequencer output is routed through a pre-trained RAVE (Realtime Audio Variational auto-Encoder) model. Specifically, the percussion.ts model was employed in this study (Acids-IRCAM, 2022).

To achieve a more natural rhythmic feel, we implemented a mechanism allowing the system to shift onsets off the rigid 16-step grid. Specifically, the Pure Data patch introduces a variable delay on off-beats to emulate the "swing" feel characteristic of human drummers. While a standard swing ratio typically assigns $2/3$ of the beat duration to the first eighth note and $1/3$ to the second, this ratio often fluctuates according to the tempo—tending toward a $1/2$ (straight) ratio at faster tempi.

In our system, this expressive timing is controlled by a single parameter labeled "entropy." The LLM is prompted to provide an entropy value between 0 and 0.2 based on the input audio descriptors. A value of 0 signifies a strictly quantized, "on-the-grid" performance, while 0.2 represents a significant "behind-the-beat" shift. The effect of this parameter is illustrated in Figure 2, which plots the onset strength relative to its temporal position in the grid. Two playback instances are compared: one with an entropy of 0 and another with a value of 0.17. The measurements show the displacement of the second onset. This hybrid approach combining contextual decision making by the AI with deterministic execution by the Pure Data sequencer minimizes LLM inference time while maintaining a layer of expressive interaction.

The provided file "Sample1.wav" in the supplemental materials demonstrates these variations with first a standard "straight" pattern; a "behind-the-beat" feel applied to the second sixteenth note at 00:13; and finally, a shift applied every fifth sixteenth note for a deconstructed groove at 00:26. This latter case creates an elastic rhythmic feel — the primary motivation for adopting the term "entropy" to describe this parameter, as it represents a departure from structured, quantized timing toward more complex temporal states.

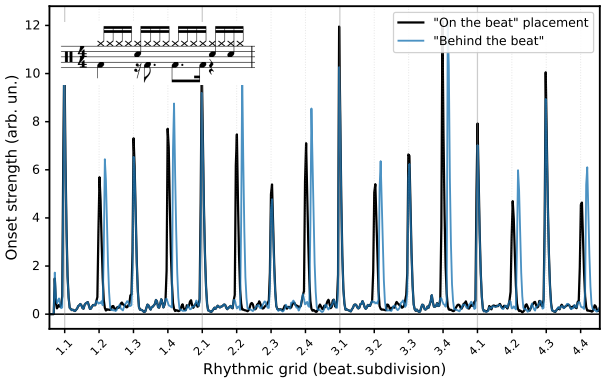

Figure 2: Onset strength versus time for the same drum pattern reproduced with two diferent time-feels, one "on the beat" (black) and the second with the odd-numbered onsets delayed "behind the beat" (blue). The magnitude of delay is controlled by the 'entropy' parameter, determined by the LLM and executed by the sequencer. This allows for the creation of micro-timing outside of the sequencer's rigid grid.

Finally, to prevent a static or monotonous auditory output, the system allows the LLM to modulate the drum kit's master volume within a range of 60% to 100%, mapped according to the input audio descriptors. This mechanism introduces a subtle dynamic contrast through a single parameter. A lower limit of 60% was intentionally selected to avoid overly discontinuous volume fluctuations as the volume setting affects the whole pattern and may be updated at every exchange with the LLM. By delegating this scaling to the Pure Data environment based on a single value provided by the LLM, we aim to achieve a preliminary implementation of a more expressive performance without introducing additional computational overhead or increasing the complexity of the prompt.

2.4 Multi-scale listening process

The system employs a dual-window analysis strategy implemented in Python via the Librosa library to process incoming audio at two distinct temporal scales. The first, referred to as the spectral window, utilizes a 3s duration to extract high-level features that capture the immediate gestural quality of the performance. These features include note density, rhythmic variation (distinguishing between metronomic stability and rubato), intensity, spectral centroid (brightness), chroma entropy (harmonic complexity), spectral flux (spectral variation), and fundamental frequency.

In parallel, a longer rhythmic window of 11 s is dedicated to tempo estimation, providing the necessary temporal depth to analyze periodicities in expressive playing. These specific descriptors were selected based on their musical relevance in the context of this study and their perceived influence on a drummer's improvisational choices. The 3 s update rate was chosen to account for the global latency of the system (whose consequences are discussed in section 3.3) and to ensure that the LLM reacts to meaningful shifts in musical energy rather than transient acoustic fluctuations, thus providing a stable foundation for the generative process.

2.5 Tempo tracking via GMM and BIC

Initial iterations of our system utilized the standard Librosa beat tracker. However, this approach exhibited significant imprecision when applied to expressive, non-metronomic human performances (Ellis, 2007; Dixon, 2001). This often resulted in erratic tempo shifts, creating a non-musical "random walk" effect in the generated percussion.

Consequently, we developed an alternative strategy for tempo detection based on temporal onset detection. By analyzing the Inter-Onset Intervals (IOI), the duration between successive note attacks, we aim to identify the fundamental IOI representing the smallest rhythmic subdivision within the analyzed window (Müller, 2015). The IOI distribution reflects the density of various rhythmic subdivisions (e.g., eighth notes, quarter notes), while its variance accounts for the performer's rhythmic imprecision. Assuming the IOI distribution is a mixture of multiple Gaussian distributions (Dixon, 2001; Peeters and Flocon-Cholet, 2012), we implemented a Gaussian Mixture Model (GMM), see Figure 3. To determine the optimal number of clusters (k) and prevent overfitting, we apply the Bayesian Information Criterion (BIC) (Schwarz, 1978; Hastie et al., 2009; Pedregosa et al., 2011):

$$BIC = k\ln(n) - 2\ln(\hat{L}) \quad (1)$$

where $\hat{L}$ is the maximum likelihood, k is the number of clusters, and n is the number of data points. The BIC operates on the principle of parsimony: while increasing k inherently improves the model's fit to the IOI distribution, it introduces a penalty for model complexity. The optimal k is thus determined by the lowest BIC value, ensuring a trade-off between likelihood and structural complexity.

Once the clusters are established, the fundamental IOI is selected by comparing Gaussian centroids; only those exhibiting integer-ratio relationships are retained as musically valid. The estimated fundamental IOI is then averaged across all consistent modes. This GMM-based approach effectively clusters similar intervals while isolating and eliminating aberrant values or performance noise. Potential tempo candidates are subsequently deduced using the formula: Tempo $= 60/(IOI_{avg} \times m)$, where $m \in \{0.25, 0.5, 1, 2, 3, 4, 6, 8\}$ and IOI_{avg} is the resulting averaged fundamental IOI. Finally, only candidates within a musically relevant range (40 to 130 BPM) are transmitted to the LLM within a structured prompt containing the "Tempo candidates".

2.6 System prompt and user prompt

The system prompt, which defines the LLM's operational constraints and creative role, is structured as follows:

```
You are an interactive AI Drummer. You are
COMPING a guitarist in real-time.

INPUT:
Every 3 seconds, you receive "AUDIO
DESCRIPTORS" and "CANDIDATE TEMPOS" of the
guitar's performance.
In particular:
- Intensity is given by RMS_dB. Below -60
means silence, the guitarist is not playing;
-35 is mezzo forte; -30 is forte; -20 is
fortissimo.
```

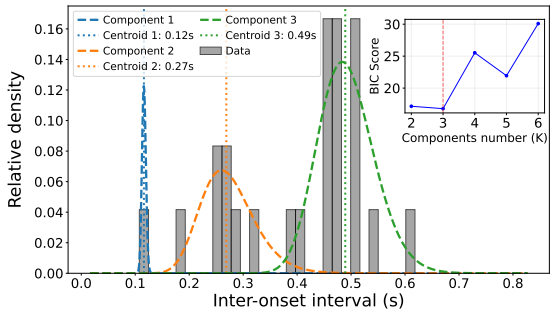

Figure 3: Relative density of inter-onset intervals (IOI) for a 10.7s audio sample. The inset displays the BIC score as a function of the number of components. The minimum score identifies an optimal model with $K = 3$ components in this example.

```
- Rhythmic_variation_coefficient reflects
rhythm accuracy. Below 0.05 means very stable;
 0.05-0.4 is standard human timing; Above 0.5
is irregular rhythm playing.
- Spectral_flux_mean varies between 0.01 and 1.
 Below 0.03 is playing calm; Above 0.07 means
nervous.
- Centropy varies between 1.8 and 2.5. Below
2.3 indicates melodic/simple intervals; Above
2.4 indicates complex chords.

TASK:
1.Analyze the guitar's density, energy, and
rhythm from the descriptors.
2.Tempo: Select a "tempo" from "CANDIDATE
TEMPOS" that matches the guitarist's flow.
3.Timing: Select a parameter "entropy" between
 0 (on the beat, punchy) and 0.2 (laid back
and relax).
4.Dynamics: Select a parameter "volume" for
drums volume between 1 (full volume) and 0.6
(60% volume) that matches guitarist's
intention.
5.Generation: Produce a 4 beats 16-step
pattern (kick, snare, hihat) IN MUSICAL
ACCORDANCE with the guitar performance.

STRICT OUTPUT RULES:
- Valid JSON only.
- REASONING: describe in one sentence how your
 kick/snare/hihat pattern reflects the guitar'
s current vibe.
- Drum pattern has 1 for sound and 0 for
silence
- Process pattern in 4 beats of 4 sixteenth
note each. For example 1000 is quarter note,
0010 is sound on second eigth note.
- Concatenate the 4 beats with a period (dot)
separator to obtain a STRICT [16 digits + 3
dots] pattern format

REQUIRED JSON STRUCTURE:
{
"reasoning": "string",
"tempo": 100,
"entropy": 0.15,
"volume" : 0.8,
"kick": "0000.0000.0000.0000",
"snare": "0000.0000.0000.0000",
"hihat": "0000.0000.0000.0000"
}
```

This prompt was developed through an iterative, empirical process aiming for musicality and optimized reliability. The LLM response is then processed in Python to filter potential errors before sending the pattern to the sequencer. For a small-to-medium scale model (8B parameters), it is essential to explicitly define the task constraints and output formatting rules. The input section provides the LLM with calibrated audio descriptors, derived from testing the guitar and soundcard signal chain.

It is important to note that the interaction is not rule-based; the LLM is not provided with a static mapping of descriptors to patterns. Instead, the model is tasked with interpreting the guitarist's musical intention and autonomously determining the tempo, entropy (micro-timing), and volume, alongside the pattern generation. This approach is intended to facilitate emergent musical behaviors that go beyond simple logic-based triggers.

A key design choice was requiring the LLM to provide a 'reasoning' string before the structured data. We observed that this approach — consistent with the Chain-of-Thought prompting technique (Wei et al., 2022) — significantly improved the model's ability to adhere to the JSON format. Furthermore, we found that requesting the 16-step pattern as a four-digit blocks separated by periods - rather than spaces or a continuous string - reduced latency and improved formatting consistency.

3 RESULTS ON THE LLM BEHAVIOUR

3.1 LLM comparison

Various Large Language Models (LLMs) were evaluated based on their inference time, instruction adherence, output integrity, and musical coherence. Five local models were installed on the workstation and assessed on their ability to generate a standard drum pattern (kick, snare, and hi-hat) in a rock style. For this benchmark we used a baseline prompt similar to the one presented in Section 2.6, though adapted to exclude audio descriptors and parameters for tempo, entropy and volume, as these are not necessary for this specific evaluation. The 'reasoning' field was also removed to exclude explanations from the output. Tests were conducted with a temperature of 0.8, and results are summarized in Table 1. Here, instruction adherence refers to the model's ability to provide the requested JSON structure without conversational filler, while output integrity denotes the precision of the 16-step rhythmic string (e.g., correct bit-count and punctuation).

Our empirical tests indicate that compact models (3B – 4B) are prone to hallucinations in this context, frequently failing to maintain the required string format for the patterns. Conversely, models at or exceeding 13B parameters introduced latencies greater than 10 seconds, which is prohibitive for near real-time interactive performance. Consequently, a balance must be struck between reasoning capabilities, stability, and execution speed. Among the tested models, Llama-3.1-8B emerged as the "sweet spot". It demonstrated high output integrity, maintained a consistent latency of approximately 1.1 s and generated musically coherent patterns with appropriate rhythmic variance.

3.2 The LLM as a contextual arbitrator

Several steps have been implemented to help the LLM to generate a musical drum pattern with a consistent tempo and to avoid hallucinations in the output of the LLM like

Table 1: Comparative performance of various LLMs for real-time drum pattern generation. For comparison, the cloud-based GPT-4o-mini model was also tested as a baseline.

LLM Model	Latency	Instruction adherence & output integrity
GPT-4o-mini (OpenAI)	Variable ($>$ 1s)	Reliable formating but requires persistent internet connection (cloud-based).
Llama-3.2-3B	$\sim$0.7s	Inconsistent sequence length ($>$ 16 steps).
Phi-3-mini (3.8B)	$\sim$1.3s	Pattern nested within conversational prose.
Llama-3.1-8B	$\sim$1.1s	Reliable formatting; rhythmic variance and musical coherence.
Llama-2-13B	$\sim$18s	Precise format, coherent but lacks pattern variation.
Phi-3-14B	$>$ 20s	Unstructured output; format instructions ignored.

undesired text or absurd values of tempo. Initially, the raw fundamental IOI was transmitted directly to the LLM, tasking the model with calculating all possible tempo candidates within a predefined range. However, we observed that while the LLM could perform these arithmetic operations in isolation, its performance degraded significantly when tasked with simultaneous rhythmic generation.

To mitigate this, our current strategy involves precalculating a list of tempo candidates in Python. The LLM is then only required to act as a contextual arbitrator, choosing from the proposed list. This design choice offloads the technical computation, allowing the model to focus exclusively on musical decisions and pattern generation, as we observed that the 8B model could not reliably handle both computational and creative tasks within the same inference iteration.

By running two listening windows in parallel, our system gives an alternative to the "octave problem" in tempo detection (the ambiguity between half-time and double-time (Schreiber and Müller, 2018)). By leaving the final arbitration to the LLM, the system selects the most appropriate tempo based on contextual descriptors from the spectral window (e.g., note density and spectral flux). This shifts tempo detection from a purely computational task to a musically informed one, where context dictates the pulse. Furthermore, this multi-scale approach addresses the inherent latency trade-off: while the 11s window facilitates a more stable tempo estimation, the 3s window allows for near-real-time gestural interaction. This enables the drum patterns remain sensitive to changes in the performer's energy while staying anchored to a longer term temporal structure.

3.3 LLM interactivity

The system is specifically designed for an aesthetic aligned with free improvisation (Bailey, 1993), rather than a soloist performing over a fixed accompaniment or groove. In the latter case, tempo fluctuations would need to be strictly minimized to prevent disruptions in the musical flow. The core objective of this study is to evaluate the ability of an LLM to translate performance descriptors into a coherent musical language and to explore the potential for interactive musical dialogue. It is important to emphasize that no explicit musical rules were provided to the LLM for the pattern generation. While the system provides a structural frame, a 4/4 measure with sixteenth-note rhythmic subdivision, the LLM is given the autonomy to fill those subdivisions or remain silent, based entirely on its pre-training.

Our primary result confirms that the integrated system is fully functional. The LLM interprets the audio descriptors and generates drum patterns alongside the requested expressive parameters. The Pure Data sequencer receives these patterns and adapts the playback parameters accordingly. Furthermore, the integration of RAVE significantly enhances the sonic experience by providing a non-repetitive and timbrally rich output. It avoids the mechanical repetitive effect typical of static sample playback, resulting in a substantially more organic and responsive rhythmic layer of sound. This shows that our system is both robust and portable on a consumer-grade workstation. An example of a performance utilizing the full system is provided in the file "Sample2.wav" in the supplemental materials (in all examples, the guitarist is the author of this study).

Evaluating the musical interaction is a complex challenge that lies beyond the primary scope of this study. However, we present here a preliminary demonstration of the system's responsiveness. To test this, the guitarist alternated between active playing and complete silence. Silence is a fundamental structural component in music, and the decision to cease playing often constitutes a significant musical event. The result of this experiment is illustrated in Figure 4, which superimposes the guitar's signal level and the drum activity over time. The guitar's signal level was measured in decibels relative to full scale (dBFS). To quantify drum activity, we calculated the sum of active triggers at each pattern (represented by '1's in the 16-step pattern) across all three instruments. Consequently, in this figure, drum activity is represented as an integer value ranging from 0 to 48.

The guitarist maintained silence during the intervals [130s-210s], [300s-345s], [405s-580s], and from 700s until the end of the session. It can observed that at 130s and 300s, the LLM mirrored this silence. In both instances, the model autonomously resumed generation after an indeterminate period. At 405s the LLM decreased its output density but only reached total silence after a significant delay, despite the continued silence of the guitar. Finally, at 700s, the LLM maintained its density showing no immediate reaction to the silence. After stopping later, it eventually resumed playing at 865s. The corresponding audio for figure 4 is provided in the file "Sample3.wav".

It is also noteworthy to examine the 'reasoning' field provided by the LLM during these periods of silence. The model generated justifications as: «The guitarist has stopped playing, maintaining steady hihat to mainain a musical feeling», «Still waiting for the guitarist to play» or

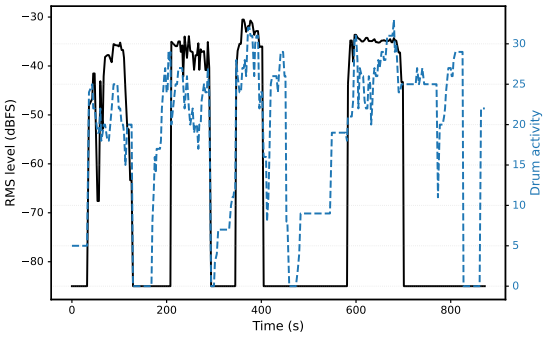

Figure 4: Superimposition of the guitar input signal (black) and the drum activity (blue/dashed) over time during an experiment where the guitarist alternates between playing and silence. The LLM activity fluctuates in response to the guitar signal, though in a non-deterministic manner, demonstrating the agent's inferential interaction rather than a fixed mapping.

«To maintain tension while anticipating their return to play, I'll keep the same neutral kick-snare pattern».

These results demonstrate an expected tendency toward responsiveness, but the reactions are notably non-deterministic. Because the LLM-based system is not rule-based, it is not programmed to react immediately or in a predefined manner. While the underlying mechanism is a probabilistic process influenced by temperature and token sampling, the resulting behaviour mimics certain human-like musical strategies, such as anticipation or the preservation of the tension. We believe this absence of rigid contraints is precisely what makes interacting with such a system compelling. Rather than a predictable tool, the LLM acts as a stochastic collaborator. Its internal "logic" remains opaque, but even so it produces emergent musical responses that suggest a level of awareness of the local context beyond simple signal-triggered automation.

Regarding the musical construction of a performance, the inherent total processing latency of approximately 3 seconds makes simultaneous musical events — such as synchronized hits — unlikely to be intentional. However, when such coincidences surprisingly occur, they can be interpreted as "happy accidents", mirroring the emergent synchronicity often found in human-to-human improvisation. In a free improvisation context, performers naturally adapt by ear, and musical transitions may sometimes unfold over several seconds or bars. Consequently, while system latency remains a significant factor, it is not entirely prohibitive for meaningful interaction. The human performer's ability to anticipate and respond to the agent's delayed output allows for a functional, albeit displaced, conversational flow.

Current limitations include the absence of global phrasing, large-scale structural construction, and the ability to transition intentionally between distinctive musical movements. Currently, certain parameters — such as the RAVE model — can be manually updated via a MIDI controller ("Sample4.wav"); however the long-term objective is for these transitions to occur autonomously. Furthermore, in this study, the LLM is provided with context from only the two previous exchanges, representing a limited temporal window relative to a full performance. We found that extending the context window further increased latency and degraded the model's adherence to the required output format. Developing a performance on a larger temporal scale would therefore require the implementation of a more sophisticated long-term memory architecture.

4 PERFORMER STUDIES

In this section, we evaluate the system through three performer studies involving professional improvisers: the author on electric guitar, a pianist and a vibraphonist. These latter two instruments were specifically selected for their percussive and rhythmic properties. The objective is to determine to what extent professional musicians can cope with the technical constraints previously described.

4.1 Public performance on electric guitar

As previously noted, the system exhibits limitations when constructing a performance over larger temporal scales (typically ten minutes). For a live execution, it was therefore necessary to introduce the capability to manually control certain system actions to create a musical narrative. The performance featured in the first part of the video demonstration took place at ICMC 2026 in Hamburg and follows a three-part structure, previously determined by the guitarist. In the first part, the system operates autonomously without manual intervention. The guitarist interacts with the agent, specifically exploring silence as a structural component. For the second part, the agent's tempo is locked at 130 BPM and the instrumentation is reduced to a single instrument, altering the ensemble's orchestral balance. Finally, in the third part, the system is set back to its original autonomous state. The addition of a "stop" button serves as a safeguard, preventing a potential lack of spontaneous interruption by the system and ensuring control over the performance's conclusion, given its non-deterministic nature.

In this manner, the guitarist was able to present a musical piece demonstrating a live, real-time dialogue between human and machine using the proposed system. The planned changes in the second part provided the necessary contrast for an engaging narrative. It should be noted that several rehearsal sessions were required for the guitarist to fully apprehend the agent's behaviour and limitations. The primary difficulties stem from the absence of direct rhythm mimicry (as the listening mode does not analyze rhythm) and the lack of immediate responsiveness. Consequently, the musician must develop alternative strategies, such as anticipating a delayed response, or the next drum hit by synchronizing with the agent's internal pulse. The performance was well received by the audience at the end.

4.2 Perfomer study with piano

The system was evaluated with a pianist having no prior experience with AI-based improvisation tools. The recording of this session is available in the second part of the video demonstration. The protocol involved providing no specific instructions to the musician apart from the existence of a multi-second response delay. After the session, his qualitative impressions were gathered.

The pianist highlighted the difficulty of constructing a musical structure with such latency, describing himself as being "already elsewhere" musically by the time the machine reacted to his intentions. He also noted a lack of sonic contrast, as the system produced a nearly constant polyphonic drum texture. Conversely, he responded positively

to the agent's playing dynamics. Note that a post-session analysis revealed a slight echo of the piano captured by the AI's audio input, which may have artificially increased its responsiveness during the session.

A second testing phase was conducted with the pianist using the pre-implemented MIDI commands, allowing a human operator (the author) to control the system's output in real-time without latency. Depending on the musical context, this enabled the operator to alternate between a polyphonic texture and a single drum (by muting the other kit elements) and to adjust the output volume, specifically facilitating the creation of silence. The pianist reported feeling significantly more comfortable in this hybrid configuration, where human intervention helped manage the AI agent's performance parameters to ensure enhanced musical coherence.

4.3 Performer study with vibraphone

Following the session with the pianist, his observations were integrated to increase the system's responsiveness while leaving the core generative task to the LLM. Heuristic rules were established and implemented in the code to act as safeguards on silence, drum sound and tempo and to dynamize the musical dialogue. These rules rely on the last five spectral windows (3 seconds each): 1) if the intensity of the two last windows falls below a predefined threshold (in dB), the drum stop playing, 2) if a window exceeds a specific intensity threshold the drum resume playing 3) if the intensity exceeds a higher maximum threshold, the RAVE pre-gain is increased, resulting in a more saturated and gritty drum sound, 4) if the note density falls below a threshold the system switches to a single drum element, 5) if the rhythmic variation coefficient is less than a threshold, indicating a musician stable tempo playing, the drum tempo is locked. This optimized version of the system was tested by a vibraphonist familiar with AI-based systems (see third part of the video demonstration). The musician particularly appreciated the rhythmic patterns generated by the LLM, which allowed for a fluid and responsive interaction. This hybrid configuration enabled him to conduct an improvisation that he judged to be fully satisfactory.

5 DISCUSSION

In this study, we explored the possibility of a musical dialogue with an LLM by proposing an architecture capable of minimizing latency for real-time interaction. The program can be used as is, as illustrated during the author's live performance (performer study number 1), although the addition of a few manual MIDI controls was necessary to compensate for the system's limited capacity for long-term structural coherence. We highlighted its capacity to interpret silence, and our performer studies demonstrate the LLM's ability to generate rhythmic proposals that enable interactive dialogue. However, further work remains necessary to evaluate the influence of other audio descriptors on the model's understanding and musical output.

The main limitation revealed by the performer studies remains its lack of responsiveness—with an overall latency of approximately 3 seconds—which necessitates a period of adaptation and specific training with the machine. The three performer studies reveal that the perception of the system (felt as either a "musician" or a "smart drum machine") is strongly influenced by the human's level of experience in interacting with artificial agents.

The system's expressivity could also be improved: the LLM frequently generates dense polyphonic patterns that can appear repetitive, although it occasionally manages to modify the instrumentation (for example, by only keeping the hi-hat) in response to the musician's playing. To avoid overloading the LLM with additional instructions, we implemented parallel heuristic rules allowing the system to quickly interrupt itself, stabilize the tempo, or momentarily filter the instrumentation based on the audio descriptor values. These rules do not generate content but dictate the system's interaction modalities. This hybrid approach introduces safeguards (or agency mediators) that increase predictability within a musical context. This makes the interaction more reliable for other performers and facilitates a smoother appropriation of the system. These adjustments provide an additional expressive dimension and enrich the performance, as illustrated by our performer studies, particularly study number 3. We are currently working on flexibilizing this mediation layer by replacing rigid, threshold-based rules with a more adaptive framework capable of integrating multiple audio descriptors simultaneously. In conclusion, while the core of musical generation and the parameters for tempo, micro-timing, and volume rely on the LLM, the addition of a mediation of agency through these heuristic rules allows for a faster onboarding with the system.

6 CONCLUSION AND PERSPECTIVES

This paper presents a system integrating a local LLM for interactive musical improvisation. By addressing the critical limitations of latency, tempo instability, and mechanical sound playback, we have developed a portable performance partner capable of engaging in a responsive dialogue with a musician. The integration of a GMM-BIC tempo tracking method provides necessary rhythmic stability, while the use of Llama-3.1-8B as a contextual arbitrator enables musically informed decision making and pattern generation derived from internal reasoning. Finally, the implementation of RAVE neural synthesis bridges the gap between symbolic AI and organic sonic expression.

The results show that an 8B model can successfully balance computational constraints with generative output, provided the system is designed to offload arithmetic tasks and focus on musical intention. Our findings highlight how non-deterministic local interactions, combined with a non rule-based algorithm, create a compelling and unpredictable environment for live performance. However, the performer studies reveal that using the system requires a period of adaptation. To address this limitation, enhance responsiveness and implement safeguards for more consistent interaction, we introduced a set of heuristic rules external to the LLM. These rules control a limited number of parameters such as the output volume of the drums or its sound, to ensure a more reliable musical dialogue.

Future work will involve a more granular analysis of the generated drum patterns to further refine the interaction logic. In addition, we aim to refine the hybrid approach by moving beyond the heuristic rules toward more flexible frameworks, such as constraint-based programming, to maintain the system's non deterministic and unpredictable nature while ensuring its consistency. The objective is to facilitate an autonomous large-scale musical narrative, allowing both the human and the machine to co-construct a performance that evolves meaningfully over time.

AUTHOR DECLARATIONS

The author reports no conflicts of interest. AI tools were used for language editing and code debugging, while all research questions, design decisions, analysis, and conclusions remain the author's own. Furthermore, the author wishes to thank Mariano Camarasa and Jeremy Friedman for their invaluable musical contributions on piano and vibraphone during the performer studies.

REFERENCES

Acids-IRCAM (2022). RAVE models download page. https://acids-ircam.github.io/rave_models_download.

Agostinelli, A., Denk, T. I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., Huang, Q., Jansen, A., Roberts, A., Tagliasacchi, M., Sharifi, M., Zeghidour, N., and Frank, C. (2023). Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*.

Assayag, G., Bloch, G., Cont, A., and Dubnov, S. (2006). Omax brothers: a dynamic topology of agents for improvisation learning.

Bailey, D. (1993). *Improvisation: Its Nature and Practice in Music*. Da Capo Press.

Caillon, A. and Esling, P. (2021). Rave: A variational autoencoder for fast and high-quality neural audio synthesis. *arXiv preprint arXiv:2111.05011*.

Dixon, S. (2001). Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58.

Ellis, D. P. W. (2007). Beat tracking with dynamic programming. *Journal of New Music Research*, 36(1):51–60.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, NY, second edition.

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A. M., Hoffman, M. D., Dinculescu, M., and Eck, D. (2019). Music transformer: Generating music with long-term structure. In *Proceedings of the International Conference on Learning Representations*.

Jambois, O., Guerra, L., D'Agostino, M. E., and Fornós, S. (2026). AI-enhanced interactive guitar improvisation system. In *Art Actors in the Age of AI*. Europia. Forthcoming.

Ji, S., Luo, J., and Yang, X. (2023). A survey on deep music generation: Multi-level representations, algorithms, and case studies. *arXiv preprint arXiv:2307.03384*.

Müller, M. (2015). *Fundamentals of Music Processing*. Springer.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peeters, G. and Flocon-Cholet, J. (2012). Perceptual tempo estimation using gmm-regression. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies (MIRUM)*, pages 45–50, New York, NY, USA.

Schreiber, H. and Müller, M. (2018). A single-step approach to musical tempo estimation using convolutional neural networks. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pages 98–105, Paris, France.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Tatar, K. and Pasquier, P. (2019). Musical agents: A typology and state of the art towards musical metacreation. *Journal of New Music Research*, 48(1):56–105.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Li, F.-F., Chi, E., Le, Q., and Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 24824–24837.