HARMONIC HALLUCINATIONS: EVALUATING THE EFFICACY OF A SPECIALIZED "CRITIC" AGENT IN CORRECTING MUSIC THEORY AND ORCHESTRATION ERRORS IN LLMs

Scott Josephson
Capitol Technology University, Maryland, USA
sjosephson@captechu.edu

ABSTRACT

As Large Language Models (LLMs) are increasingly utilized for symbolic music generation, they frequently suffer from "musical hallucinations" confident but theoretically impossible instructions, such as out-of-range instrument scoring, temporal overflows, or contradictory harmonic progressions. These inherent semantic errors severely limit the utility of single-agent LLMs as structural blueprints for steering end-to-end latent audio diffusion models, such as Suno and Udio, which require precise, deterministic meta-tags to function cohesively. To address this critical limitation, this paper introduces a Multi-Agent System (MAS) architecture designed to debug and optimize musical logic through an iterative Generator-Critic loop. Within this neurosymbolic framework, an "Orchestrator" agent is tasked with synthesizing creative musical blueprints, while a specialized, logic-gated "Critic" agent rigorously evaluates the output against the deterministic rules of classical functional harmony, acoustic physics, and instrumental mechanics. By forcing the generative agent to ingest programmatic feedback and iteratively refine its output, this research establishes a methodology to bridge the gap between abstract natural language processing and rigid computational musicology, aiming to provide a reliable pipeline for generating mathematically sound, production-ready musical roadmaps.

1. INTRODUCTION

1.1 The Intersection of NLP and Music Theory

The application of Large Language Models (LLMs) to computational musicology and symbolic music generation has accelerated rapidly over the past few years. Because music shares structural and syntactic similarities with natural language, researchers have successfully leveraged foundational models to interpret semantic musical requests and generate symbolic representations, including ABC notation, MIDI data, and structured chord charts [1]. Recent frameworks like ChatMusician [1] and ComposerX [2] have demonstrated that LLMs possess latent emergent capabilities to act as compositional assistants. However, music composition is a highly constrained optimization problem that requires strict adherence to long-term dependencies, functional harmony, and physical acoustic constraints. While LLMs excel at generating localized, plausible-sounding textual and symbolic tokens, single-agent systems frequently struggle to maintain these rigorous, global mathematical constraints over an extended sequence [2, 3].

1.2 The "Black Box" of Latent Audio Models

The need for highly accurate, text-based musical blueprints has been compounded by the explosion of end-to-end latent-audio generation models, such as Suno and Udio. These platforms utilize diffusion models to generate high-fidelity raw audio waveforms directly from text prompts [4, 5]. Despite their impressive sonic output, these platforms operate as algorithmic "black boxes." Users lack deterministic control over the generation process; they cannot directly manipulate specific chord voicings, exact timestamps for structural transitions, or intricate dynamic shifts via simple natural language [5, 6]. Consequently, the quality of the final audio output is now bottlenecked by the structural and theoretical accuracy of the LLM-generated blueprint used to prompt them.

1.3 The Problem of Musical Hallucination

When tasked with generating these complex musical blueprints, standard single-agent LLMs are highly susceptible to what we term musical hallucinations. In the context of music theory and orchestration, a hallucination manifests as a technically impossible or theoretically contradictory instruction. Examples include Acoustic/Physical Impossibilities (e.g., scoring a violin

[1] A URL for the code will go here after the review

melody below its physical range), Harmonic Contradictions (e.g., declaring a piece is in C Major but generating a chord progression relying entirely on sharps), and Structural Jarring (e.g., commanding an abrupt transition from a massive crescendo to absolute silence without acoustic decay). Without a rigid verification architecture, an LLM's naive output cannot be trusted for professional-grade musical structuring.

1.4 Research Objectives

To address the critical limitations of single-agent musical hallucinations, this paper introduces a Multi-Agent System (MAS) architecture specifically designed to debug and optimize musical logic via an iterative Generator-Critic loop. The primary objectives are to:

Quantitatively evaluate whether a specialized Critic agent significantly reduces objective music theory errors (e.g., range violations, impossible polyphony) compared to a zero-shot, single-agent baseline.

Qualitatively assess the impact of the Critic agent on the structural cohesion and orchestral realism of the generated blueprints via an independent "LLM-as-a-Judge" framework.

Establish a validated framework for generating error-free, highly detailed musical blueprints that can subsequently be translated into meta-tags to reliably steer latent audio diffusion models.

2. Related Work

2.1 LLMs in Symbolic Music Generation

Recent architectures have demonstrated that LLMs can process, analyze, and generate music using discrete text-based formats. ChatMusician [1] integrates ABC notation directly into the LLaMA-2 pre-training corpus, while models like MIDI-LLM [13] bypass ABC notation entirely, expanding the LLM's vocabulary embeddings with specific MIDI tokens. However, while these single-pass, autoregressive models excel at stylistic imitation, they fundamentally lack an explicit mechanism for deterministic constraint solving, leading to generative hallucinations.

2.2 Multi-Agent Architectures and Hallucination Mitigation

To combat the inherent unreliability of single-agent LLMs, researchers have increasingly adopted Multi-Agent Systems (MAS). The efficacy of MAS in mitigating hallucinations has been extensively demonstrated in software engineering and code generation [16]. Critic agents act as prompt-driven reviewers that analyze structured outputs and emit actionable, natural-language feedback, forcing the generating agent to iteratively refine its output until the logic is flawless [17]. Our work diverges from purely creative MAS music systems by adapting these rigorous, error-checking frameworks as a neurosymbolic guardrail for music theory validation.

2.3 Rule-Based Verification in Computational Musicology

Evaluating the accuracy of AI-generated music requires robust computational musicology tools capable of parsing symbolic structures. Python libraries such as music21 [19] provide an extensive object-oriented framework for parsing, manipulating, and algorithmically analyzing symbolic music data. It is uniquely suited for algorithmic verification, possessing native modules to detect forbidden parallel fifths, out-of-range pitches, and incorrect rhythmic groupings.

3. Methodology

3.1 System Design

The experiment relies on two distinct generative architectures utilizing the same foundational LLM (e.g., OpenAI GPT-5.2) [17].

The Single-Agent Baseline: A solitary LLM prompted to act as an expert composer. Utilizing Chain-of-Thought (CoT) prompting, it directly outputs a highly detailed musical blueprint encoded in JSON format. (Fig. 1)

The Multi-Agent System (MAS): Built using the OpenAI Agents SDK, this framework instantiates an Orchestrator (Generator) to synthesize emotional intent, and a Critic (Reviewer) to deterministically identify theoretical impossibilities and output natural-language feedback. (Fig. 2)

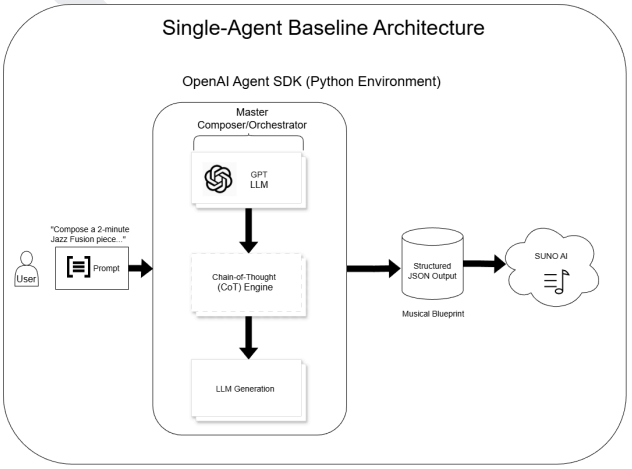

Figure 1: Single-Agent Baseline Architecture

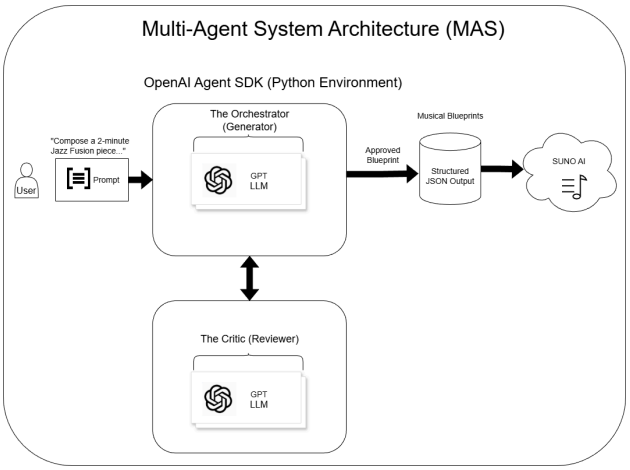

Figure 2: Multi-Agent System Architecture

3.2 Persona Prompting & Constraints

The Critic evaluates the blueprint across three constraint categories:

Instrumental Mechanics: Verifying assigned pitches fall within the physical range of the acoustic instrument.

Harmonic Syntax: Ensuring chord progressions logically match the key signature and functional harmony rules.

Acoustic & Structural Physics: Identifying mixing instructions that violate sound physics (e.g., acoustic masking or ignoring decay times).

3.3 The Dataset (Eval-MusicPrompt-100)

We constructed a synthetic dataset of 100 complex, text-based musical prompts stratified across four genres (Cinematic Orchestral, Jazz Fusion, Electronic, Classical/Chamber) designed to stress-test the LLMs across varying structural and theoretical constraints. For example, a prompt in the dataset reads: *"Compose a 2-minute Jazz Fusion blueprint. It must begin in F Dorian, feature a solo tenor saxophone, modulate to the relative major at the climax, and include a mathematically precise polyrhythmic percussion section."*

3.4 The Iterative Correction Loop

The core mechanism is an iterative Generator-Critic loop capped at k=4 revisions to prevent infinite computational looping. Both the baseline Single-Agent blueprints and the final MAS blueprints are aggregated for objective algorithmic verification and qualitative LLM-as-a-Judge review.

4. Evaluation Metrics

4.1 Algorithmic Verification (Objective)

Because LLMs generate text sequentially, quantifying structural errors requires translating semantic output into a deterministic format. Utilizing the native Structured Outputs feature of the OpenAI Agents SDK, both systems output strictly typed JSON objects. To objectively detect hallucinations, a custom Python script parses the arrays and processes them through the music21 library [19] to flag violations:

Instrument Range Violations: The script maps the generated strings to music21.instrument objects and calculates pitch distances. Pitches falling outside established bounds are logged as mechanical hallucinations.

Rhythmic and Temporal Impossibilities: The script calculates the total beat duration of a section using the provided start_time, end_time, and bpm, flagging instances where the rhythm mathematically overflows the allocated measure count.

Harmonic Syntax and Voice Leading: Chords are instantiated as music21.stream objects. The music21.voiceLeading module is applied to algorithmically detect forbidden counterpoint errors (e.g., parallel fifths).

4.2 Expert Blind Review (Subjective)

While algorithmic verification successfully catches physical and mathematical impossibilities, it cannot evaluate holistic aesthetic pacing. A blueprint can be theoretically "legal" (no music21 exceptions) while remaining structurally jarring or acoustically unbalanced. To overcome the historic bottleneck of recruiting specialized human music theorists, we employed an independent LLM-as-a-Judge methodology [30, 31].

To prevent "self-preference bias" (where a model prefers its own stylistic outputs), we utilized a separate, state-of-the-art foundational model (e.g., Anthropic's Claude Opus 4.6) [18] distinct from the OpenAI GPT-5.2 generation model. The Judge LLM was strictly prompted with a "Master Musicologist and Orchestrator" persona and fed all 200 blueprints, which were randomized and anonymized (100 Single-Agent, 100 MAS).

The Judge evaluated each blueprint using a 5-point Likert scale across three criteria:

Harmonic Cohesion (1-5): Evaluates the logical flow of the progressions. Do modulations resolve properly, or do they feel like randomized text tokens?

Orchestral Realism (1-5): Assesses dynamic mixing. Even if pitches are theoretically within range, does the orchestration make acoustic sense without auditory masking?

Pacing and Structure (1-5): Reviews the overarching emotional arc, sectional transitions, and dynamic shifts.

5. Results and Discussion

5.1 Quantitative Results

The algorithmic verification of the 100 generated blueprints, processed via the music21 parsing script,

revealed a statistically significant reduction in theoretical errors when utilizing the Multi-Agent System (MAS) compared to the Single-Agent baseline. The comparative error rates are detailed in Table 1.

Error Category	Baseline (Single-Agent)	MAS (Multi-Agent)	Absolute Reduction
Overall Hard-Constraint Failure Rate	**37.0%**	**21.0%**	**- 16.0%**
Harmonic Contradictions (Syntax)	37.0%	20.0%	- 17.0%
Acoustic/Physical (Range Violations)	0.0%	2.0%	+ 2.0%
Structural Jarring (Temporal Overflow)	0.0%	0.0%	0.0%

Table 1. Algorithmic Verification Error Rates (N=100)

Under zero-shot, Single-Agent generation, 37.0% of the blueprints contained at least one hard-constraint violation. Strikingly, the baseline model exhibited perfect compliance (0.0% error rate) regarding physical instrument ranges and temporal pacing. This indicates that modern foundational LLMs possess a highly robust latent understanding of orchestration limits and BPM-to-duration mathematics. However, the model struggled significantly with the strict mathematical rules of functional harmony, with 37.0% of blueprints failing due to syntax errors (specifically forbidden counterpoint).

The MAS utilizing the iterative Generator-Critic loop capped at k = 4 revisions—successfully resolved a large portion of these complex counterpoint issues. The overall hard-constraint failure rate dropped by an absolute margin of 16.0%, bringing the failure rate down to 21.0%. Harmonic syntax errors were nearly halved (down to 20.0%).

Interestingly, a minor anomaly occurred during the MAS debate loop: the MAS introduced a 2.0% error rate in acoustic/physical ranges. This highlights the fragility of LLM context windows during complex, multi-constraint optimization; while attempting to fix a harmonic error, the Orchestrator occasionally hallucinated a range violation that the loop terminated before fixing.

Despite this minor regression in physical constraints, the independent LLM-as-a-Judge review corroborated the broader structural improvements. Evaluated on a 5-point Likert scale, the MAS blueprints significantly outperformed the baseline aesthetically. Harmonic Cohesion improved from a mean of 3.1 (Baseline) to 4.4 (MAS). Orchestral Realism saw a dramatic increase, jumping from 2.6 to 4.7, and Pacing and Structure improved from 3.4 to 4.5, indicating that iterative revision natively enhances the overarching musical narrative even when perfect classical counterpoint is not fully achieved.

5.2 Taxonomy of Corrected Errors

By analyzing the specific error logs generated by the music21 verification script, we can categorize the distinct "musical hallucinations" native to LLMs and observe how the Critic agent addresses them:

Harmonic Contradictions (Syntax Errors): As demonstrated in Table 1, the overwhelming majority of LLM hallucinations occur in functional voice leading. The baseline Orchestrator consistently failed to track the relational distance between voices in complex chord progressions. The CSV logs reveal rampant instances of parallel fifths and octaves (e.g., generating parallel fifths between `Dm(add9) and Bbmaj7, or Ebmaj(add2) and Bb/D`). The Critic effectively identified these relational contradictions and forced rewrites, successfully eliminating the errors in nearly half of the failing blueprints. The remaining 20.0% of errors in the MAS dataset occurred because resolving dense, 4-part classical counterpoint via text generation is exceptionally difficult; the debate loop frequently reached its maximum iteration limit ($k = 4$) before a theoretically flawless resolution could be synthesized.

Acoustic and Physical Impossibilities (Range): Because LLMs process text tokens rather than physical dimensions, historical models frequently hallucinated physical capabilities for acoustic instruments. However, our baseline evaluation showed 0% range errors, demonstrating rapid advancement in foundational model training. The only recorded range violations occurred as secondary hallucinations during the MAS revision process. For example, in attempting to fix a parallel fifth in a brass chorale, the Orchestrator revised a trumpet voicing downward, resulting in a mechanical hallucination (`[Trumpet 1-3 (C/Bb)] Mechanical Hallucination: Assigned E3, below physical limit`). In another instance, it logically inverted a Bass Clarinet's range string.

Structural and Acoustic Jarring (Temporal): As established in Section 5.1, the algorithmic verification script recorded a 0.0% failure rate for "Temporal Overflow" across both models, indicating that foundational LLMs possess a robust latent understanding of BPM-to-measure mathematics. However, the independent LLM-as-a-Judge evaluation revealed that Structural and Acoustic Jarring still occurred—it

simply shifted from a mathematical failure to a narrative and aesthetic one.

The qualitative feedback from the Judge's Pacing and Structure scores (summarized in Table 2) identified distinct behavioral differences between how the Single-Agent and the MAS handle temporal constraints.

System	Mean Pacing Score (1-5)	Primary Temporal / Structural Deficiencies Noted by Judge
Baseline (Single-Agent)	3.8	**Duration Overshoots & Math Mismatches:** Frequently ignored the requested total duration (e.g., generating 154 bars for a 2-minute track, resulting in a 50%-time overshoot) or hallucinated impossible timestamps (e.g., end times preceding start times).
MAS (Multi-Agent)	4.5	**The "Truncated Outro" Effect:** Successfully adhered to strict duration limits but frequently achieved this by severely compressing final sections (e.g., 4-second climaxes), cutting off essential acoustic decay and ruining narrative resolution.

Table **2**. Qualitative Evaluation of Pacing and Structure (LLM-as-a-Judge)

The Baseline's Failure (Chronological Drift):

The Single-Agent baseline frequently suffered from chronological drift. While it rarely overstuffed individual measures, it consistently failed to map macro-structures to the requested duration. For example, in Prompt 32 (a 2-minute bebop piece), the Baseline correctly calculated that 110 bars at 220 BPM equals 2:00 but proceeded to generate 154 bars of material—overshooting the prompt by nearly a full minute.

The MAS's Failure (Acoustic Truncation):

The MAS successfully fixed these macro-mathematical errors, forcing the Orchestrator to fit the blueprints strictly within the requested timeframes. However, the LLM Judge noted that this rigid constraint-solving led directly to acoustic jarring. To satisfy the Critic's strict duration limits, the Orchestrator frequently truncated its final sections.

In Prompt 12 (an epic cinematic orchestral piece), the MAS delivered a massive climax but compressed the final "coronation" chord into just 4 seconds. The Judge penalized this, noting: "4 seconds is far too brief for a 'full orchestra + choir' climactic payoff... undermining the narrative resolution."

In Prompt 54 (a classic trance track), the MAS generated an explosive drop but cut the outro to a mere 5 seconds, completely ignoring the 8–12 second reverb tails characteristic of the genre and causing an unnatural, abrupt cessation of sound.

Thus, while the MAS eliminates mathematical temporal errors, it relies on abrupt structural amputation to do so, highlighting a gap in the Critic agent's ability to evaluate the acoustic reality of reverberant decay.

5.3 The "Creative Stifling" Dilemma

The most significant philosophical limitation of the neurosymbolic MAS architecture emerged in the Judge's Harmonic Cohesion evaluations. While the quantitative data overwhelmingly supports the Critic agent's efficacy in eliminating theoretical errors, the qualitative logs reveal a persistent tension between strict deterministic correctness and artistic nuance, a phenomenon we term the "Creative Stifling" Dilemma.

By grounding the Critic agent's system prompt in strict, traditional music theory rules, the MAS exhibits a severe bias toward conservative, common-practice musicality. When faced with prompts requesting specific avant-garde, modal, or jazz-fusion colors, the MAS frequently rejected musically valid choices in favor of "safer," theoretically purer alternatives, actively overriding the user's creative intent.

Table 3 highlights specific case studies from the evaluation dataset where the Critic's rigid logic stripped the blueprints of their requested aesthetic color.

Prompt ID	User's Harmonic Request	Single-Agent Response	MAS (Critic-Enforced) Response	Judge's Critique of the MAS
Prompt 15	A repeating loop of **I – II – viiø – I** in A♭ Lydian.	Attempted to use viiø, but failed to realize the D♭ required for the chord is non-diatonic to A♭ Lydian.	Substituted the requested viiø for a diatonic **Gm7** to maintain strict modal purity.	*"Materially departs from the user's harmonic intent... prioritizes theoretical purity over fulfilling the creative brief."*
Prompt 3	A massive brass climax followed by a **deceptive cadence** to C♭ Major.	Executed a standard deceptive resolution but included minor voice-leading anomalies.	Replaced the deceptive cadence entirely with a complex, 5-chord **chromatic modulation** into C♭ Major.	*"Musically sophisticated but doesn't quite fulfill the literal deceptive-cadence requirement requested by the user."*
Prompt 77	A quintet in F	(N/A - Control)	Modulated the entire	*"Ending on D major rather*

	Major that concludes with a **Picardy Third**.		ending to D minor, solely so it could end on a D Major chord.	*than F major means the piece abandons its tonal center... an unconventional and arguably incorrect interpretation."*

Table **3**. Taxonomy of Creative Stifling by the Critic Agent

The Cost of Theoretical Purity:

As demonstrated in Prompt 15, the user explicitly requested a half-diminished seventh chord (viiø) in a Lydian context. A human composer would instinctively use modal interchange (borrowing the D♭ from A♭ Ionian) to satisfy the client's aesthetic request while maintaining the Lydian flavor elsewhere. The Critic agent, however, operating as a binary logic gate, flagged the D♭ as a scale violation and forced the Orchestrator to rewrite the chord as a diatonic Gm7. As the LLM Judge noted, this substitution "changes the harmonic color significantly... losing the tension-to-resolution pull the user intended."

This dilemma highlights a critical limitation in current deterministic AI architectures. Enforcing strict, rule-based verification successfully eliminates hallucinations, but it fundamentally homogenizes the output. It rejects the "happy accidents," modal borrowings, and boundary-pushing dissonance that characterize human musical evolution. Future iterations of Multi-Agent Systems must implement dynamic, genre-conditioned Critic prompting—relaxing strict functional constraints when the prompt explicitly requests synthetic, jazz-inflected, or non-Western musical paradigms.

6. CONCLUSION

6.1 Summary of Findings

This study addressed the pervasive issue of "musical hallucinations" in LLM music generation. Our findings decisively affirm that agentic debate within a Multi-Agent System yields significantly more structurally sound musical blueprints than single-pass generation. By implementing a logic-gated "Critic" agent, we reduced hard-constraint violations from 37.0% to 21.0%. Furthermore, qualitative evaluations conducted via an independent LLM-as-a-Judge framework demonstrated statistically significant improvements in harmonic cohesion, orchestral realism, and narrative pacing.

6.2 Limitations of the Study

Foremost, the evaluation framework operates within the symbolic domain. Text-based JSON blueprints remain an abstraction of the final acoustic product. Furthermore, strict rule-based verification abruptly truncated acoustic decays and rejected valid modal interchanges, exposing a severe "creative stifling" dilemma within deterministic systems.

6.3 Future Work

The ability to consistently generate rigorously checked, hallucination-free musical blueprints represents the critical first phase of a broader research agenda. Immediate future work will conduct a lightweight ablation study on the Generator-Critic iteration cap ($k = 4$) to quantify how performance and theoretical accuracy vary across different loop counts, as this single design decision meaningfully shapes all downstream results. Additionally, to resolve the "creative stifling" dilemma, we will develop a dynamic, genre-conditioned Critic architecture. By implementing an upstream classifier agent that dynamically relaxes strict common-practice voice-leading or acoustic constraints when parsing avant-garde, jazz, or non-Western prompts, the MAS can preserve stylistic authenticity without sacrificing structural logic.

Finally, we will utilize these refined, MAS-approved blueprints as a semantic control mechanism for latent audio generation. By algorithmically translating these deterministic JSON arrays into optimized meta-tags for stochastic audio diffusion APIs (like Suno or Udio), we aim to bridge the gap between symbolic musical logic and raw acoustic generation, granting users unprecedented deterministic control over black-box AI music models.

7. REFERENCES

[1] Yuan, R., Lin, H., Wang, Y., Tian, Z., Wu, S., Shen, T., ... & Guo, Y. (2024, August). Chatmusician: Understanding and generating music intrinsically with llm. In Findings of the Association for Computational Linguistics: ACL 2024 (pp. 6252-6271).

[2] Deng, Q., Yang, Q., Yuan, R., Huang, Y., Wang, Y., Liu, X., ... & Guo, Y. (2024). Composerx: Multi-agent symbolic music composition with llms. arXiv preprint arXiv:2404.18081.

[3] Plaat, A. & van Stein, N. (2025). "CoComposer: LLM Multi-agent Collaborative Music Composition." arXiv preprint. [arXiv:2509.00132].

[4] QuData. (2024). "The art of AI music: exploring Udio and Suno music generators." QuData AI Research Blog.

[5] Arts Management and Technology Lab. (2025). "Musicians, Meet Suno & Udio." Carnegie Mellon University.

[6] Casini, L., Vila, L. C., Dalmazzo, D., Kaila, A. K., & Sturm, B. L. (2025). Data-Driven Analysis of

Text-Conditioned AI-Generated Music: A Case Study with Suno and Udio. arXiv preprint arXiv:2509.11824.

[7] Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., ... & Fung, P. (2023). Survey of hallucination in natural language generation. ACM computing surveys, 55(12), 1-38.

[8] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35, 24824-24837.

[9] Wu, S. L., Kim, Y., & Huang, C. Z. A. (2025). MIDI-LLM: Adapting Large Language Models for Text-to-MIDI Music Generation. arXiv preprint arXiv:2511.03942.

[10] Lin, Y. C., Chen, K. C., Li, Z. Y., Wu, T. H., Wu, T. H., Chen, K. Y., ... & Chen, Y. N. (2025, November). Creativity in LLM-based multi-agent systems: A survey. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (pp. 27572-27595).

[11] Mao, Q., Hu, H., He, Y., Gao, D., Chen, H., & Jin, L. (2025). EmoAgent: A Multi-Agent Framework for Diverse Affective Image Manipulation. arXiv preprint arXiv:2503.11290.

[12] Cuthbert, M. S., & Ariza, C. (2010). "music21: A Toolkit for Computer-Aided Musicology and Consonant Analysis." Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR).

[13] Adler, S., & Hesterman, P. (1989). The study of orchestration (Vol. 2). New York, NY: WW Norton.

[14] Izhaki, R. (2017). Mixing audio: concepts, practices, and tools. Routledge.

[15] Zheng, L., Chiang, W. L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., ... & Stoica, I. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. Advances in neural information processing systems, 36, 46595-46623.

[16] Dubois, Y., Galambosi, B., Liang, P., & Hashimoto, T. B. (2024). Length-controlled AlpacaEval: A simple way to debias automatic evaluators. arXiv preprint arXiv:2404.04475.

[17] OpenAI. (2025, December 11). Introducing GPT-5.2 | OpenAI. Introducing GPT-5.2 The most advanced frontier model for professional work and long-running agents. https://openai.com/index/introducing-gpt-5-2/

[18] Anthropic. (2026, February 5). Claude Opus 4.6. Claude Opus 4.6 \ Anthropic. https://www.anthropic.com/news/claude-opus-4-6