MIDI-PHOR: Multi-View Distillation for Music Understanding and Captioning

Steven Au
Independent Researcher
steventinwing@gmail.com

Abstract

A central limitation of current music understanding frameworks is the reliance on audio embeddings, which frequently yields interpretations lacking traceable ties to explicit musical elements such as notes, dynamics, and instrumentation. We address this gap with MIDI-PHOR, a MIDI-first framework that converts symbolic data into structured, queryable representations for reasoning. MIDI-PHOR distills each piece into three complementary views: a symbolic view capturing pitch, meter, and key; a time-series (TS) view that tracks rhythmic salience, texture, and role activity; and an instrument-role graph encoding ensemble interactions. With evidence-linked claims, experiments demonstrate reduced hallucinations compared to raw-MIDI baselines and offer a robust, auditable bridge between symbolic data and semantic music understanding.

1 Introduction

Music understanding is a foundational challenge in multimodal artificial intelligence, requiring systems to not only classify audio but to analyze, describe, and reason about complex structural relationships. While natural language processing has achieved remarkable interpretability through Large Language Models (LLMs), progress in music AI has been dominated by generative objectives. Systems such as MusicLM (Agostinelli et al., 2023) and MusicGen (Copet et al., 2024) focus primarily on learning probabilistic audio-text correspondences for synthesis. While these models excel at creative generation, they function as "black boxes" regarding musical structure, lacking the ability to provide traceable, verifiable justification for their outputs (e.g., citing specific notes, harmonic progressions, or orchestration techniques).

Symbolic data formats like MIDI offer a promising alternative by encoding music as discrete events—pitch, rhythm, instrumentation, and control changes—enabling analysis at a granular level. However, two significant barriers have hindered symbolic music understanding. First, symbolic data lacks the implicit timbral cues (e.g., specific synth textures or production effects) that standard models rely on to infer genre and mood. Second, existing attempts to bridge this gap, such as MIDI-Caps (Melechovsky et al., 2024), often rely on rendering MIDI to audio to extract features. We argue that this approach is flawed: it introduces non-deterministic variance based on the synthesizer or SoundFont used, confusing the model with artifacts that do not exist in the underlying composition.

We introduce MIDI-PHOR (**P**olyphonic **H**armony **O**rchestration **R**easoner), a framework that targets an explicit grounded reasoning layer with three queryable views. MIDI-PHOR treats MIDI not merely as a sequence of tokens, but as a structured knowledge object for music understanding rather than synthesis. To overcome the semantic gap without relying on unstable audio rendering, we distill MIDI into three complementary views:

1. A **Symbolic View** that captures ground-truth pitch, meter, key regions, and harmonic progression;
2. A **Time-Series View** that projects discrete tokens into continuous signal proxies (e.g., "symbolic spectral flux," "groove entropy") to represent texture and dynamics natively;
3. An **Orchestration Graph** that encodes ensemble roles and interactions between instruments(e.g., *locking*, *call-and-response*) as explicit edge relations.

We demonstrate that this multi-view representation significantly reduces hallucination compared to raw-MIDI baselines and offers a robust, invariant bridge between symbolic data and semantic description. Our core contribution is the introduction of a grounded reasoning layer that separates

Proceedings of the 4th Workshop on NLP for Music and Audio (NLP4MusA 2026), pages 33–43
March 28, 2026 ©2026 Association for Computational Linguistics

language generation from musical truth verification. By distilling MIDI into a queryable relational backend, we provide a representation that constrains LLMs to descriptive claims strictly supported by MIDI-level evidence.

2 Related Work

Audio-Language Models. The field has bifurcated into generation and retrieval. Generative systems like MusicLM (Agostinelli et al., 2023) and MusicGen (Copet et al., 2024) utilize vast audio-text pairs to synthesize music, but they function as "black boxes" lacking explicit structural reasoning. Conversely, joint embedding models like CLAP (Elizalde et al., 2022) and MuLan (Huang et al., 2022) align audio and text in a shared latent space for zero-shot retrieval. While effective for global semantics (e.g., "happy jazz"), these embeddings often fail to capture granular musical details such as specific chord inversions or instrument roles. To address data scarcity, LP-MusicCaps (Doh et al., 2023) and MusicLingo (Deng et al., 2024) leverage LLMs to generate pseudo-captions, yet these remain bound by the subjectivity of the underlying audio analysis—a limitation highlighted by the Chordify Annotator Subjectivity Dataset, which shows only $\sim$54% expert agreement on complex harmony (Koops et al., 2019).

Symbolic Music Understanding. Symbolic data offers a path to objective analysis. Early pretraining approaches like MusicBERT (Zeng et al., 2021) applied masked language modeling to MIDI to learn structural representations, while recent foundation models like Moonbeam (Guo and Dixon, 2025) improve this with domain-aware tokenization. However, these models are primarily designed for sequence classification or generation, not interpretability. The closest precedent to our work is MIDI-Caps (Melechovsky et al., 2024), which introduced a large-scale MIDI-caption dataset. However, as shown in Table 1, MIDI-Caps and similar frameworks largely rely on *rendering* MIDI to audio to extract timbre and genre features. This dependency re-introduces the variance of audio synthesis (soundfonts, reverb) into the pipeline. MIDI-PHOR diverges from this by proposing a symbolic-only bridge, deriving texture and timbral proxies directly from token distributions to ensure invariant, auditable reasoning.

3 Methodology

We refit the music captioning task as a grounded reasoning problem, where the LLM functions as a claim selector over a structured musical knowledge base. To enable this, we construct a unified, bar-aligned representation of musical data designed for arrangement, structure, and symbolic reasoning. The pipeline consists of three view-specific extractors—(i) symbolic analysis, (ii) symbolic time-series projection, and (iii) orchestration graph construction—whose outputs are materialized in a relational DuckDB backend. This design makes every high-level caption claim traceable to explicit musical evidence (notes, tracks, bars, chords, and derived bar-level signals). We then distill these tables into LLM-friendly artifacts (ScoreSpec, ScoreSpec-Lite, enhanced facts, hierarchical facts) used for captioning and retrieval as shown in Figure 1.

To facilitate this architectural design, which ensures every high-level descriptive claim is traceable to explicit musical evidence, we define the following data hierarchy:

- *View*: One of the three analytical perspectives used to interpret the MIDI source.
- *Fact*: An atomic, bar-indexed entry in the relational backend.
- *Claim*: A natural language statement produced by the LLM describing a musical phenomenon.
- *Evidence Pointer*: A metadata link anchoring a generated claim to its corresponding fact in the database.

We distill these relational tables into four hierarchical artifacts—ScoreSpec, ScoreSpec-Lite, enhanced facts, and hierarchical facts—used for captioning and retrieval as shown in Figure 1. By using these artifacts, we effectively transform the task into a form of grounded extractive summarization, ensuring the generation process is strictly conditioned on retrieved facts rather than the latent priors of the language model. Detailed specifications for these artifacts are provided in Appendix A.1

3.1 Symbolic View Analysis

We process MIDI files using the robust event parsing of `miditoolkit` and `pretty_midi` to populate normalized tracks and notes tables, aligned to a bar backbone derived from the MIDI time-signature maps. For harmonic analysis, we leverage the musicological theory engine of `music21` to perform chordification, generating Roman-numeral annota-

Table 1: Comparison of current captioning systems by modality and output granularity. **TS** denotes Symbolic Time-Series proxies; **MIR** stands for Music Information Retrieval. **Claims** refers to evidence-linked descriptive statements.

System	Modality	Signal	Output(s)	Global	Section
LP-MusicCaps	Audio	Pseudo-captions	Caption	✓	
MusiLingo	Audio	Captioning	Caption	✓	
FUTGA	Audio	Temporal	5 captions	✓	✓
SonicVerse	Audio	Temporal/MIR	Long captions	✓	✓
MidiCaps	MIDI	Symbolic + Audio	Captions	✓	
MIDI-PHOR (Ours)	MIDI	Sym + TS + Graph	Claims + Evidence	✓	✓

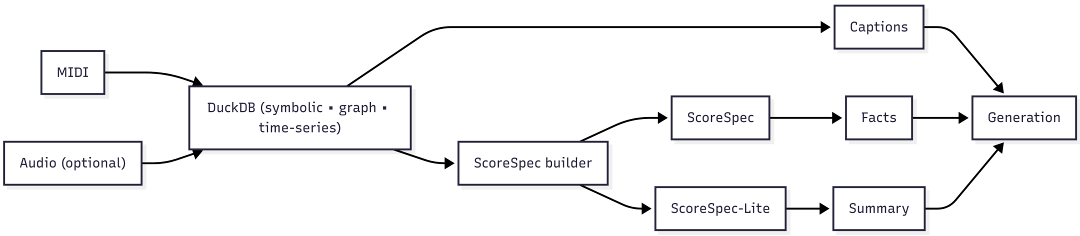

Figure 1: Pipeline from MIDI/audio to ScoreSpec, facts, and generation.

tions (rn) for harmonic summaries and deterministic cadence detection.

To handle modulation without destabilizing Roman-numeral labeling, we separate global and local key regions. A single global key estimate is stored in `key_changes` as the reference for Roman numerals, while a windowed tonal-center tracker stores time-varying local keys in `key_regions` (bar spans with confidence). This preserves consistency for RN-based reasoning while enabling analysis of local tonal shifts.

To ensure robustness to tempo changes, core arrangement statistics are bar-indexed rather than time-indexed. In particular, polyphony is computed on a tick-aligned grid inside each bar (yielding invariance to tempo scaling), while density is computed as normalized note-on activity per bar.

3.2 Symbolic Time-Series Projection

A key goal of MIDI-PHOR is to avoid dependence on rendered audio for structural cues. While prior work renders MIDI to audio and then applies signal processing for segmentation and tag prediction, synthesized audio introduces confounds tied to the chosen soundfont and rendering settings. Instead, we project discrete symbolic events into bar-level signals stored in `ts_bar`, enabling robust structure reasoning without waveform processing.

We compute pitch-class distributions (bar-level chroma) and derive a symbolic novelty signal (`novelty_bar`) from the distance between consecutive chroma vectors. We also compute repetition measures (`repeat_score_bar`, `recurrence_density_bar`) based on cosine similarity between bar chroma vectors. Rhythmic salience is estimated using onset entropy on a quantized 16th-note grid (`onset_entropy_16th`), alongside bar-level syncopation and backbeat proxies derived directly from symbolic onsets. These signals support deterministic structural event extraction (e.g., cadence peaks from Roman-numeral transitions) and enable section boundary discovery using novelty peaks.

3.3 Orchestration Graph Construction

To represent ensemble structure beyond token sequences, we construct a heterogeneous orchestration graph. Nodes represent tracks and sections, and edges represent interactions inferred from symbolic evidence. For example, a `RHYTHMIC_LOCK` relation is created when bass and drums exhibit strong co-activity patterns over multiple bars. We also infer accompaniment-style relations (e.g., `SUPPORTS_HARMONY_OF`) when pad/comp roles co-occur with melody across sustained spans, and `CALL_RESPONSE` patterns via bar-level alternation of track activity. Each edge is stored with evidence spans in bar units (`edge_evidence`), enabling auditable statements such as “the pad supports the melody” grounded in explicit bar ranges.

Table 2: Invariance checks ($N = 100$). Pass rate denotes the fraction of tracks where structural boundaries remain stable (Jaccard ≥ 0.9) under transformation.

Transformation	**Pass**	**Expected Behavior**
Transpose (±12 semi)	0.929	Harmonic content rotates.
Tempo (×0.8–1.25)	0.929	Tick-based features remain.
Velocity (×0.5–1.5)	0.929	Dynamics shift predictably.

Table 3: Factuality evaluation (Claim–Evidence verification, $N = 97$). MIDI-PHOR significantly reduces hallucinations compared to baselines.

Method	**Claim Cov. ↑**	**Evid. Cov. ↑**	**Verify Pass ↑**	**Halluc. Rate ↓**
MIDI-PHOR (Ours)	0.718	**0.990**	**0.804**	**0.186**
Symbolic-Only	**1.000**	0.974	0.701	0.317
Render+Audio	**1.000**	0.954	0.486	0.538

3.4 Distillation and Integration

The DuckDB representation is distilled into three complementary LLM-facing formats: (i) a section-aware scaffold (merged sections with active roles and salient signals), (ii) an enhanced flat fact list for prompt injection, and (iii) a hierarchical fact index organizing facts by domain (harmony, rhythm, orchestration, motifs). These are serialized through a standardized SCORESPEC JSON interface, ensuring that any generated caption is strictly conditioned on retrieved, auditable facts rather than latent priors.

4 Evaluation

We evaluate MIDI-PHOR across three dimensions: (i) feature reliability via invariance testing, (ii) factual grounding via claim verification, and (iii) computational efficiency. Experiments utilize a random sample of 100 tracks from the Clean Lakh MIDI dataset (Raffel, 2016).

4.1 Reliability via Invariances

Lacking large-scale ground truth for structural events (e.g., section boundaries), we use *invariance checks* as a proxy for robustness. We apply controlled perturbations—transposition, tempo scaling, and velocity scaling—to the input MIDI and measure the stability of detected section boundaries. A trial is considered a pass if the Jaccard similarity between the original and perturbed boundary sets is ≥ 0.9. Table 2 confirms that MIDI-PHOR's symbolic extractors remain stable under standard musical transformations, whereas audio-based features often drift due to synthesis artifacts.

4.2 Factuality and Hallucination

To quantify grounding, we employ a *Claim–Evidence* verification protocol. The model generates a caption as a JSON list of claims, where each claim includes a value and a specific evidence pointer (e.g., a bar range). We verify these claims against the underlying database.

We compare MIDI-PHOR against two baselines: (1) *Symbolic-Only* (standard LLM prompting with raw MIDI tokens), and (2) *Render+Audio* (synthesized audio with MIR-based taggers, mimicking MIDI-Caps). As shown in Table 3, MIDI-PHOR achieves the highest verification pass rate (0.804) and the lowest hallucination rate (0.186). The *Render+Audio* baseline struggles with hallucination (0.538), primarily because the LLM generates descriptions unsupported by the specific synthesis used. See Appendix **??** for baseline specifications

Note that claim coverage should be interpreted under a fixed claim budget rather than as absolute completeness.[1] Within this budget, MIDI-PHOR utilizes available slots most effectively (0.990 evidence coverage).

4.3 Prompt Sensitivity

To assess stability, we generated captions using three distinct prompts on a 30-song subset. The set of retrieved slots remained consistent (Jaccard similarity 0.769). When a specific slot was claimed, the value agreed across prompts 84.3% of the time, with perfect consistency in evidence pointers. This indicates that MIDI-PHOR's extracted view effectively constrains the LLM to the underlying data regardless of phrasing changes.

4.4 Computational Profile

We report end-to-end runtime normalized by song duration. MIDI-PHOR averages 2.36 s/min on a standard CPU, noticeably outperforming the Render+Audio baseline (3.53 s/min). This performance gap validates our hypothesis regarding the limitations of audio-based approaches: the baseline

---

[1] We cap the number of emitted claims to K=10 (`max_claims`) to keep outputs concise and comparable across methods, bound token costs, and prevent trivial coverage inflation from listing every available slot. Under this budget, claim coverage reflects how effectively a method uses a fixed claim budget, not how many total facts exist in the representation.

is dominated by the synthesis step (2.72 s/min), which introduces significant external dependencies and disk overhead. By bypassing this bottleneck, MIDI-PHOR not only achieves faster processing but also maintains a self-contained pipeline with a direct, inspectable link from symbolic events to the generated evidence.

5 Conclusion

We have introduced MIDI-PHOR, a framework that addresses the lack of interpretability in music understanding by reframing captioning as a grounded reasoning task. By distilling MIDI into a structured knowledge base comprising a symbolic scaffold, a synthesis-free time-series projection, and an orchestration graph, we provide a representation that is both machine-readable and human-auditable.

Our approach departs from "black-box" audio-text mapping to prioritize explicit reasoning over grounded evidence. The evaluation results validate this architectural choice: MIDI-PHOR demonstrates robustness to musical transformations where audio features frequently degrade, while achieving a significant reduction in hallucination rates (18.6% vs. 53.8% for audio-based baselines). By forcing the generation process to cite specific bars, tracks, and graph edges, we ensure that claims about harmony, texture, and interplay are strictly traceable to the source data. This framework offers a practical path toward trusted music AI, where descriptive outputs are not just plausible, but verifiable.

Limitations

Lack of Ground Truth. The primary limitation of this work is the scarcity of large-scale, human-annotated ground truth for the structural phenomena we extract (e.g., specific *drop* locations, *call-and-response* segments). Consequently, our evaluation relies on internal consistency checks—such as invariance testing and claim-evidence verification—rather than precision/recall metrics against a gold standard. While invariance confirms that our detectors are stable, it does not strictly prove their perceptual validity to a human listener.

Absence of Vocal and Lyrical Content. Standard MIDI files encode instrumental performance data but inherently lack audio vocals and lyrical text. Consequently, MIDI-PHOR cannot extract features related to vocal timbre or thematic content, which are often central to high-level musical understanding. While MIDI can encode a vocal melody line symbolically, recovering the full semantic context represents a challenge intrinsic to the format. Addressing this limitation would necessitate the development of fusion models that integrate symbolic representations with aligned audio or textual modalities.

Heuristic Nature and Data Dependency. Several graph edge detectors (e.g., `RHYTHMIC_LOCK`) rely on threshold-based heuristics. While tuned for pop/rock structures in the Lakh MIDI Dataset, these heuristics may require recalibration for genres with different rhythmic conventions, such as Jazz or Classical. Furthermore, the system is bound by the quality of the input symbolic data; MIDI files with missing percussion tracks, extreme quantization, or malformed events will inevitably degrade the quality of the extracted features and the resulting captions.

Ethics Statement

Subjectivity and Affective Labeling. Music description is inherently subjective. A core ethical risk in automated captioning is the presentation of affective labels (e.g., *sad*, *triumphant*) or stylistic interpretations as objective facts. MIDI-PHOR is designed to mitigate this by prioritizing structural evidence over emotive language. However, users should remain aware that even symbolic analysis can encode bias; for example, the definition of *complexity* or *consonance* is culturally specific. Generated captions should be interpreted as evidence-based descriptions of the file content, not authoritative musicological or emotional analysis.

Bias in Symbolic Corpora. Symbolic datasets like the Lakh MIDI Dataset predominantly represent Western popular music traditions. Consequently, the extracted features—particularly those related to harmony (Roman Numerals) and meter (4/4 backbeat)—may under-represent or mischaracterize music from non-Western traditions or avant-garde traditions that do not adhere to standard tonal and metric grids. We position MIDI-PHOR as a complementary tool to audio-first approaches, acknowledging that it is most effective within the specific cultural contexts where MIDI is a standard representation.

References

Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. 2023. Musiclm: Generating music from text. *Preprint*, arXiv:2301.11325.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. 2024. Simple and controllable music generation. *Preprint*, arXiv:2306.05284.

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

SeungHeon Doh, Keunwoo Choi, Jongpil Lee, and Juhan Nam. 2023. Lp-musiccaps: Llm-based pseudo music captioning. *Preprint*, arXiv:2307.16372.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 2022. Clap: Learning audio concepts from natural language supervision. *Preprint*, arXiv:2206.04769.

Zixun Guo and Simon Dixon. 2025. Moonbeam: A midi foundation model using both absolute and relative music attributes. *Preprint*, arXiv:2505.15559.

Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. 2022. Mulan: A joint embedding of music audio and natural language. *Preprint*, arXiv:2208.12415.

Hendrik Vincent Koops, W. Bas de Haas, John Ashley Burgoyne, Jeroen Bransen, Anna Kent-Muller, and Anja Volk. 2019. Annotator subjectivity in harmony annotations of popular music. *Journal of New Music Research*, 48(3):232–252.

Jan Melechovsky, Abhinaba Roy, and Dorien Herremans. 2024. Midicaps: A large-scale MIDI dataset with text captions. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR 2024)*.

Colin Raffel. 2016. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. Ph.D. thesis, Columbia University.

Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu. 2021. MusicBERT: Symbolic music understanding with large-scale pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 791–800, Online. Association for Computational Linguistics.

A Supplementary Results

This appendix consolidates evaluation details and qualitative examples that are referenced but not expanded in the main paper.

A.1 Artifact Design and Reproducibility

To ensure reproducibility and provide a template for the captioning task, we extract analysis at four abstraction levels. These artifacts were designed to distill every note and control change from the granular MIDI event stream into a structured, queryable JSON representation.

- **ScoreSpec**: Encodes fine-grained musical data, including segments, instrument metadata, pitch-class spans, and motif occurrences, alongside a complete structural graph.
- **ScoreSpec-Lite**: A section-level summary (tempo, bar ranges, density, active instruments, and harmonic summaries) designed for rapid inspection and lower token overhead.
- **Enhanced Facts**: A human-readable bullet list of salient features generated from structured summaries for direct prompt injection.
- **Hierarchical Facts**: Reorganizes findings into analytic categories (structure, harmony, rhythm, orchestration, motifs, and form) to support topic-focused reporting and precise querying.

A.2 LLM Auditing Metrics and Prompt Sensitivity

We introduce a suite of auditing metrics to measure the structural integrity of the generated captions, verifying them against the ground-truth database rather than relying solely on surface-level text overlap. Table 4 details these results. The *Evidence coverage* score of 0.990 shows that the model consistently cites specific bar ranges or track IDs, allowing users to trace statements back to the source signal. *Claim coverage* (0.718) reflects the fraction of available database slots that appear in the final output. This value remains below 1.0 because the system encourages concise captions capped at a fixed claim budget, prioritizing readability over exhaustive enumeration. Despite this selectivity, the *Verification pass rate* of 0.804 and low *Hallucination rate* of 0.186 demonstrate that the model maintains high factual accuracy for the claims it generates.

We further assessed the robustness of the model's grounding by testing three prompt variations on a subset of 30 songs. The *Slot-set stability* score of 0.769 indicates that changing the prompt effectively steers the model to select different features, such as focusing on harmony versus rhythm. Even when the topical focus shifts, the underlying

Table 4: LLM auditing metrics (measured). Values are computed by matching generated claim slots and evidence pointers to extracted database facts.

Metric	**Value**	**Interpretation**
Claim coverage	0.718	Fraction of expected slots that the model attempts to fill
Evidence coverage	0.990	Fraction of claimed slots that include a valid evidence pointer
Verification pass rate	0.804	Fraction of claimed slots whose values match DB ground truth
Hallucination rate	0.186	Fraction of claimed slots that fail verification
Slot-set stability (Jaccard)	0.769	Prompt sensitivity: overlap of which slots are claimed (30 songs, 3 prompts)
Value agreement	0.843	Prompt sensitivity: agreement of claimed values when the slot is claimed
Evidence agreement	1.000	Prompt sensitivity: agreement of evidence pointers under prompt variants

reasoning remains deterministic. The perfect *Evidence agreement* of 1.000 confirms that whenever different prompts lead the model to cite the same feature, they point to the exact same evidence in the database.

B Extended Experimental Protocols

B.1 Baseline Architecture and Audio Optionality

We compare Midi-Phor against a MidiCaps-style hybrid baseline, which renders MIDI to audio to extract timbral and genre features. We reproduced this experimental setup to align with prior frameworks and to quantify the impact of synthesis-induced variance on grounding.

A key distinction in our implementation is the reduction of **dependency friction**. The original audio-based extraction requires heavy external libraries (e.g., C++-backed audio processors) that are often difficult to configure. In contrast, Midi-Phor functions as a self-contained NLP pipeline. While we treat audio rendering as optional to ensure invariant reasoning, our architecture remains flexible: it supports audio conversion modules if multimodal fusion is required, offering a superset of the baseline's capabilities without enforcing its toolchain overhead.

B.2 Computational Profile and Amortization

We benchmark the end-to-end computational cost of both systems on a standard CPU setup. Midi-Phor (symbolic-only) averages 2.40 s/min ($p_{50} = 2.22$), whereas the baseline averages 3.53 s/min.

The primary difference lies in the **synthesis bottleneck**. MIDI→WAV rendering dominates the baseline pipeline, accounting for approximately 77% of total runtime (2.72 s out of 3.53 s) and introducing significant disk usage. Furthermore, the two costs serve different ends:

- **One-off vs. Reusable:** The baseline primarily incurs preprocessing costs to produce disposable features for captioning.
- **Amortized Extraction:** Midi-Phor materializes reusable, queryable tables in `DuckDB`. Once computed, these artifacts support multiple downstream tasks (retrieval, QA, analysis) without re-processing the source MIDI.

B.3 Measuring Reliability via Jaccard Invariance

Given the scarcity of purely symbolic ground truth, we sought a rigorous metric to evaluate the stability of the system's reasoning. We employ Jaccard similarity (Intersection over Union) to measure the consistency of detected section boundaries under controlled perturbations, such as transposition and tempo scaling.

In this context, Jaccard similarity serves as a proxy for **"hallucination resistance."** If the system's structural understanding (e.g., identifying a Chorus) shifts significantly due to a simple key change, the model is effectively hallucinating new musical facts. Midi-Phor maintains a robust 0.929 pass rate, demonstrating that it anchors its claims in invariant symbolic relationships rather than fragile audio artifacts.

C Feature Extraction Details

This appendix details the extraction logic for the three views used in Midi-Phor. A summary of all extracted features is provided in Table 6.

C.1 Symbolic Feature Logic

Tempo-Invariant Density. Standard note density (notes per second) is sensitive to tempo changes. To ensure our features describe the *composition* rather than the *performance speed*, we calculate

Table 5: Compute profile (measured). We compare MIDI-PHOR (symbolic-only) against a MIDICAPS-style reproduction (which renders MIDI → WAV audio → MTG taggers + Essentia chords). Runtime is reported as seconds per minute of music across songs.

	Runtime (s/min)				
Stage	**Mean**	**P50**	**P95**	**Disk**	**Notes**
MIDI-PHOR					
`symbolic.run`	2.241	2.075	3.768	Low	Parsing + bar grid + symbolic features
`merge_for_song`	0.005	0.005	0.010	Low	Section merge via symbolic novelty
`graph_ext.run`	0.155	0.135	0.277	Low	Interaction graph w/ evidence spans
Total	**2.402**	**2.218**	**4.059**	**Low**	End-to-end (no audio rendering)
MIDICAPS-style					
Audio Render	2.716	2.645	3.357	High	Synthesis dominates runtime
MIDI parse (meta)	0.054	0.051	0.095	Low	Tempo/key/instrument parsing
MTG taggers	0.659	0.640	0.803	Low	Mood/genre from audio embeddings
Chords (Essentia)	0.020	0.019	0.023	Low	Chord extraction from audio
Total	**3.529**	**3.485**	**4.351**	**High**	Audio-driven pipeline

`polyphony_max` and `density_note` using absolute MIDI ticks.

Key Regions. While global key estimation is useful for high-level metadata, it fails to capture modulations (e.g., a bridge in the relative minor). We employ the Krumhansl-Schmuckler key-finding algorithm over a sliding 4-bar window to populate the `key_regions` table. This allows specific sections to be tagged with their local tonal center while preserving a stable global key for Roman Numeral analysis.

C.2 Time-Series Proxies (Symbolic Signals)

To avoid the computational cost and synthesis artifacts of audio rendering, we derive continuous signals directly from symbolic tokens.

- **Symbolic Flux:** We approximate spectral flux by combining velocity variance (ΔV) and pitch-class distribution shifts (ΔC) between consecutive bars. Sudden spikes in this signal robustly indicate section boundaries.
- **Groove Entropy:** To measure rhythmic complexity without audio, we project onsets onto a quantized 16th-note grid and compute the Shannon entropy of the resulting distribution. High entropy correlates with syncopated or "busy" rhythms, while low entropy indicates straight quantization.
- **Cadence Strength:** We calculate a heuristic "resolution score" that peaks when a Dominant (V) harmony resolves to a Tonic (I) on a strong metric downbeat.

C.3 Graph Interaction Logic

Interaction edges are inferred through rule-based heuristics applied to the track activity curves. Figure 2 shows an arrangement summary to visualize a view of sections and roles.

- **Rhythmic Lock:** Defined as a high Intersection-over-Union (IoU > 0.6) of onset locations between a Bass track and a Drum track over a 4-bar window.
- **Call and Response:** Detected by calculating the rolling correlation coefficient between the amplitude envelopes of two melodic tracks. A strong negative correlation ($r < -0.5$) implies an alternating "call and response" pattern.
- **Support:** A directed edge established when a "Pad" role's sustained duration covers $> 80\%$ of a "Melody" role's active timeline.

D Schema and Prompt Examples

We contrast the sparse input context available to the comparison method (MidiCaps) versus the rich structural context provided by MIDI-PHOR in Figure 3. To isolate the contribution of the input representation (static tags vs. temporal graph)).

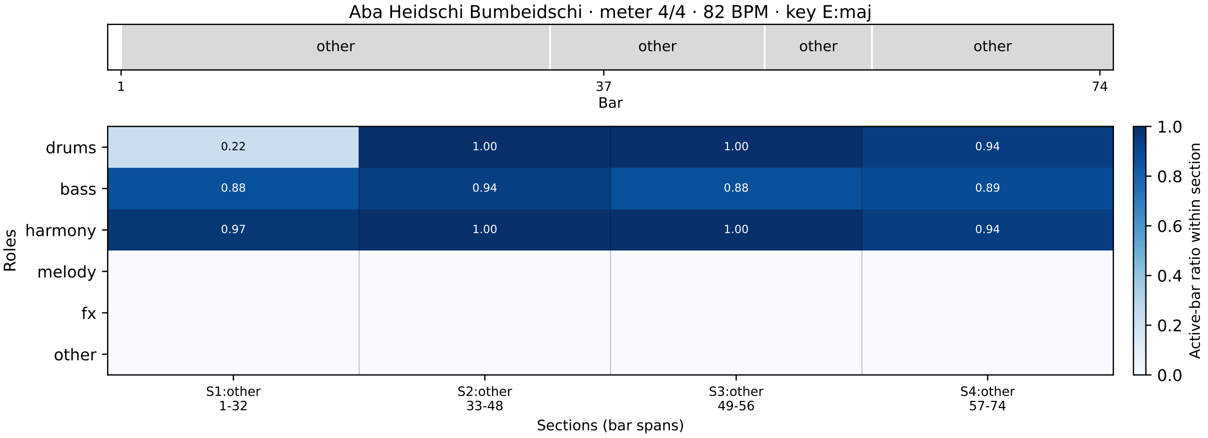

Figure 2: Arrangement view for *Aba Heidschi Bumbeidschi*. Top: merged section timeline over bars. Bottom: section×role activity heatmap (active-bar ratio per role), summarizing how drums/bass/harmony/melody enter and persist across sections. This visualization is derived purely from MIDI notes and the merged section boundaries and serves as an interpretable companion to the full interaction graph.

Table 6: Comprehensive list of features extracted by MIDI-PHOR across all three views.

Category	**Feature Name**	**Description**
Symbolic	`density_note`	Count of note onsets per bar.
	`density_rhythm`	Ratio of occupied 16th-note grid positions.
	`polyphony_max`	Max simultaneous voices (computed in ticks).
	`pitch_range`	Semitone distance between lowest/highest pitch.
	`key_local`	Windowed key estimate (4-bar sliding window).
	`harmony_roman`	Roman numeral relative to local key.
	`motif_ngram`	Top-5 recurring interval-duration sequences.
Time-Series	`flux_sym`	Proxy for spectral change (ΔVelocity $+$ ΔChroma).
	`groove_entropy`	Shannon entropy of quantized onset grid.
	`role_activity`	Normalized density curves per instrument role.
	`cadence_strength`	Heuristic score for V$\to$I resolution on downbeats.
	`novelty_bar`	Self-similarity peak detection for boundaries.
Graph	`RHYTHMIC_LOCK`	High onset IoU between Bass and Drums.
	`CALL_RESPONSE`	Anti-correlated activity envelopes between tracks.
	`SUPPORT`	Sustained Pad coverage of Melodic lines.
	`DOUBLES`	High pitch/onset identity ($> 90\%$) between tracks.

Table 7: Side-by-side qualitative comparison for *Aba Heidschi Bumbeidschi*. We contrast the generated free-form caption with the intermediate structural representation to demonstrate explicit slot filling and prompt grounding.

Midi-Phor (Ours)	**MidiCaps Baseline**
Generated Free-form Caption	
A laid-back groove anchors this track at a relaxed 82 BPM, with drums and layered textures creating a warm, full sound. The harmonic movement cycles between a bright major tonality and a darker, flattened seventh chord that adds subtle tension. The interplay between the main rhythm and punchy accents keeps the energy grounded.	A serene and uplifting composition in E major that blends orchestral and electronic elements beautifully. The gentle harp and lush string ensemble create a warm, melodic foundation, while ethereal pads and synth strings add a modern, meditative quality.
Intermediate Structure	
4/4 major key composition at 82 BPM in E major with I-bVII7#2 progression. Average polyphony of 11.6 voices, density 9.92, and syncopation 0.24. Weak backbeat (0.19). Orchestrated with harp, synth strings, steel guitar, string ensemble, and acoustic bass.	Section in E major at 82 BPM, 4/4 meter. Chord palette: B, Eb, Gb, E, D. Instrumentation: Orchestral Harp, Synth Strings 1, Acoustic Guitar (steel), String Ensemble 1, Acoustic Bass.
Grounded Claims & Evidence	**Detected Tags**
• `meter = 4/4` • `tempo_bpm = 82` • `key = E:maj` • **`progression = I-bVII7#2`** • `chord_summ = [[“Eb”, “Gb”], 8]` • **`polyphony = 11.61`** • **`density = 9.92`**	• `tempo_bpm = 82` • `meter = 4/4` • `key = E:maj` • `chord_palette = [“B”...]` • `instruments = [“Harp”...]`

MidiCaps Comparison (Tag-based)

System Prompt

I will give you paired data of tags extracted from songs and the original song captions. The task is to generate the caption from the tags.

Input Representation (Sourced from Claude 3.5 Opus Dataset)

- `Genre:` [['alternative', 'electronic'], [0.24, 0.24]]
- `Mood:` [['meditative', 'dark', 'relaxing'], [0.11, 0.11, 0.10]]
- `Instruments:` [Piano, Electric Bass, Drums]
- `Time Signature:` 4/4 `Tempo:` [125, Fast]
- `Key:` A major
- `Chords:` [['A', 'E7', 'A7', 'D'], 3]

Generated Caption (gpt-4o-mini)

A piano plays a lead melody with chords in the background while being accompanied by a bass guitar and string ensemble. The song evokes a sense of zen, darkness and has a meditative feeling to it. The time signature is 4/4 and the tempo is fast. It's in the key of A major and the main chord progression is A, E7, A7, D.

Midi-Phor Ours (Graph-based)

System Prompt

Write a concise, human-friendly music caption. Use 2 sentences for short; 3–4 for medium. Avoid jargon; describe feel, groove, and harmonic motion.

Input Context (Structured Analysis – Abridged)

- **Summary:** Length: $\sim$188s (94 bars) | Tempo: 120 BPM
- **Form & Texture:**
  - M1 [1-8]: density low; active: Vocal, ElecGtr... | roles comp:2, bass:1
  - M2 [9-16]: density high; active: Vocal, ElecGtr...
  - M11 [81-94]: density low; active: Vocal, Rhythm&SE
- **Harmony & Relationships:**
  - M1: Fourth B$\times$180 $\rightarrow$ C#m$\times$90
  - DOUBLES: Vocal $\rightarrow$ SteelGtr [6–94]
  - RHYTHMIC_LOCK: Bass $\rightarrow$ Rhythm&SE [6–94]

Generated Caption (gpt-4o-mini)

This energetic pop track features a steady 120 BPM groove, driven by strong backbeats and a rich texture of bass, steel guitar, and rhythmic accents. The harmonic motion shifts between minor and major chords, creating an engaging tension that invites movement. With varying dynamics across its sections, the song maintains a vibrant energy that keeps listeners captivated.

Figure 3: Vertical comparison of prompt contexts. The MidiCaps comparison (top) uses the dataset output from the Claude 3.5 Opus generation. MIDI-PHOR (bottom) ingests a serialized graph including section-level texture, harmonic rhythm, and instrument interaction. Both captions were generated using `gpt-4o-mini` to strictly compare the input representations.