UC Merced
Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Towards a Cognitive Pattern Language for Tonal Music Representation

Permalink
https://escholarship.org/uc/item/6zz3k506

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 48(0)

Authors
Monnet, Joris
Rohrmeier, Martin Alois

Publication Date
2026

Copyright Information
This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Towards a Cognitive Pattern Language for Tonal Music Representation

Joris Monnet (joris.monnet@epfl.ch)[1] **& Martin Rohrmeier**[1]
[1]Digital and Cognitive Musicology Lab, EPFL

Abstract

We present a symbolic, rule-based framework for modeling tonal music composition as the construction and transformation of structured cognitive programs. Building on the Language of Thought hypothesis, our approach treats musical scores as externally observable artifacts that encode hierarchical relations, melodic patterns, and harmonic scaffolds in a reusable symbolic system. Using Minimum Description Length (MDL) principles, we show that tonal abstractions support compression: themes function as shared models, allowing variations to be represented with fewer parameters and transformations, reflecting cognitively plausible reuse and systematicity. We demonstrate the framework on Paganini's Caprice No. 24 and Steve Reich's Clapping Music, showing how surface diversity emerges from a compact set of symbolic operations applied over inherited structures. Unlike data-driven neural models, our method preserves interpretable, inspectable representations aligned with musicological analysis. This work provides a proof-of-concept that tonal music can be formalized as an executable symbolic system, offering both a computational model of composition and a window into the cognitive structures underlying musical creativity.

Keywords: Symbolic cognition; Tonal music; Musical patterns; Computational modeling; Algorithmic composition; Abstraction; Compression; Musical representation

Introduction

A central topic in cognitive science characterizes cognition as the manipulation of structured symbolic representations by rule-governed mental programs (Fodor, 1975; Jackendoff, 2002; Newell and Simon, 1972). Within this framework, cognitive systems are understood as productive and systematic: a finite set of symbols and operations gives a large range of structured thoughts and behaviors. This perspective has been fruitfully applied to domains such as planning, problem solving, language, visual narrative. or comics and visual storytelling (Cohn, 2013, 2020).

Music cognition has a rich literature on perception, expectation, grouping, and affect (Deutsch, 1999; Huron, 2008). Yet music making, particularly tonal composition, is rarely examined as symbolic program execution. Generative models like the Generative Theory of Tonal Music (Lerdahl, 2005; Lerdahl and Jackendoff, 1985) focus on perceptual organization and analysis rather than composition as executable symbolic processes. More recent frameworks, including *Pytakt* (Nishimura and Marui, 2025), CommonMusic (Taube, 1991, Tymoczko's work (Tymoczko, 2006, 2023) and more specifically his Arca language (Tymoczko, 2024), formalize musical structure computationally. Here, we explicitly treat tonal composition as constructed and transforming symbolic programs, emphasizing representational adequacy, reuse, and description-length minimization as cognitively motivated principles.

Musical scores offer an opportunity for cognitive modeling. They are not encodings of acoustic outcomes, but structured symbolics patterns that specify actions, constraints, and transformations that can be learned, taught, analyzed, and systematically understood. Tonal music, in particular, relies on a symbolic ontology: scales, harmonic functions, patterns, and transformations. These regularities are not arbitrary: they support learnability, stylistic coherence, and productive reuse between works and composers (Gjerdingen, 2007; Temperley, 2004). From a cognitive perspective, this suggests that tonal music reflects an implicit domain-specific symbolic system, similar to a programming language internalized by composers and performers.

This view aligns naturally with the Language of Thought (LoT) hypothesis, which states that cognition operates over structured representations built from a finite set of primitives and combinatorial rules (Fodor, 1975; Newell, 1994; Pylyshyn, 1984). Such representations support systematicity, productivity, and compositionality (Fodor and McLaughlin, 1990; Jackendoff, 2002). Contemporary implementations highlight description length, abstraction, and symbolic reuse as explanatory principles in concept learning, causal reasoning, program induction, and visual representation (Chater and Vitányi, 2003; Dechter et al., 2013; Ellis et al., 2018; Lake et al., 2017; Solomonoff, 1964). Here, LoT is adopted as a representational hypothesis: if tonal music is produced and understood via symbolic cognitive structures, it should admit compact, compositional, and reusable formal descriptions.

We present a rule-based framework modeling tonal music generation as interactions of symbolic musical objects and compositional operations. Grounded in established musicological taxonomies, it represents musical material as reusable patterns that can be repeated, transformed, overlaid, and concatenated. We evaluate it on diverse examples, including Steve Reich's Clapping Music and Paganini's Caprice No. 24. Using Minimum Description Length (MDL) (P. Grünwald and Roos, 2019; P. D. Grünwald, 2007), we show that generating scores via this framework yields more compressed representations than note-by-note generation. When a theme

In G. Boleda, I. Dautriche, M. Maldonado, C. Saldana and T. Torrent (Eds.), *Proceedings of the 48th Annual Conference of the Cognitive Science Society* ©2026 the author(s). This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY).

exists, variations can be encoded with shorter symbolic descriptions than independent representations, approximating Kolmogorov compression (Chaitin, 1966; Kolmogorov, 1965; Li and Vitányi, 2008) and formally supporting the role of tonal abstractions in compact generative representation.

Unlike data-driven deep learning approaches (Hörnel and Menzel, 1998; Johnson et al., 2017), our framework is fully rule-based and symbolic. Neural models, while effective, often obscure how musical objects, structure, and relations are represented. Our focus is not surface-level generation quality but preserving explicit, interpretable symbolic information aligned with musicological concepts.

This proof-of-concept demonstrates that tonal music can be modeled as an executable symbolic system consistent with cognitive principles of systematicity, productivity, and parsimony. Beyond generating music, the framework serves as a cognitive model, capturing the structure of symbolic objects and operations that musicians reason about, manipulate, and transmit, providing a foundation for further empirical and theoretical work on symbolic musical cognition.

Model

Pattern

A *Pattern* is a container for representing and manipulating sequences of musical events. It behaves like an ordered sequence while supporting musically meaningful structural and transformational operations. A *Pattern* may contain Notes, Rests, Chords, or other event objects.

A *Pattern* may be rhythmic or non-rhythmic. In rhythmic patterns, events have explicit onset and duration, enabling time-based transformations and analyses. In non-rhythmic patterns, events are represented only by sequential ordering.

The *Pattern* abstraction is extensible and can accommodate additional event types (e.g., dynamics or articulations) without modifying its core interface.

$$\begin{aligned} P ::= \texttt{base} \mid \texttt{shift}(P, \delta) \mid \texttt{retrograde}(P) \\ \mid \texttt{concatenate}(P, P) \mid \texttt{overlay}(P, P) \end{aligned} \tag{1}$$

These production rules define a compositional language over *Pattern* expressions. From a Language of Thought perspective, the goal is not to enumerate all musically useful operations, but to specify a small set of cognitively plausible primitives from which more complex structures can be compositionally derived. In our framework, this grammar defines the hypothesis space over which structural descriptions are evaluated under a Minimum Description Length criterion.

The terminal symbol `base` denotes an atomic pattern, i.e., a finite sequence of musical events. The operator $\texttt{shift}(P, \delta)$ translates all events in P by an offset δ while preserving internal structure, and $\texttt{retrograde}(P)$ reverses event order. Sequential composition is captured by $\texttt{concatenate}(P_1, P_2)$, which appends P_2 after P_1 with appropriate time alignment, whereas $\texttt{overlay}(P_1, P_2)$ merges both patterns without altering internal temporal relations.

The auxiliary operation $\texttt{split}(P, \alpha)$ partitions a pattern into two sub-patterns (P_1, P_2), where α denotes either a temporal boundary or an index.

These operations correspond to elementary transformations over structured sequences: displacement, reversal, sequential composition, superposition, and segmentation. In LoT, cognition relies on structured symbolic representations compositionally generated from a finite set of primitives (Fodor, 1975). Our proposal follows this view by using a minimal set of domain-specific primitives sufficient to generate richer musical structures.

Some musically useful operations are not primitive but compositionally defined. Let $\Delta(P)$ denote the duration of a rhythmic pattern P. Then

$$\texttt{repeat}(P, 2) = \texttt{concatenate}\big(P, \texttt{shift}(P, \Delta(P))\big),$$

with the general case obtained recursively.

For example, Steve Reich's *Clapping Music* can be generated using the *Pattern* abstraction together with *TimeSpan* objects representing rhythmic onsets, durations, and *Rests*. The composition has attracted interest from both mathematicians (Haack, 1991, 1998) and musicologists (Sethares and Toussaint, 2015) because of its apparently simple notated structure and rhythmically complex perceptual effect.

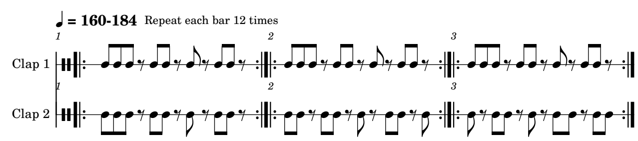

Figure 1: First 3 bars of *Clapping music* by Steve Reich

According to Reich, 2002 himself, the piece consists of a single rhythmic pattern continuously repeated by the first performer, while the second gradually shifts the same pattern forward by an eighth note every n measures (n=2or12 depending on the version), as partially shown in Figure 1. This produces a systematic phase-shifting effect that progressively obscures the beat while continuously renewing the surface texture.

Within our framework, a base *Pattern* is defined using *TimeSpan* (1) and *Rest* (0) objects of equal durations. Successive *ROL* operations then provide a compact computational reconstruction of the piece, as shown in Algorithm 1.

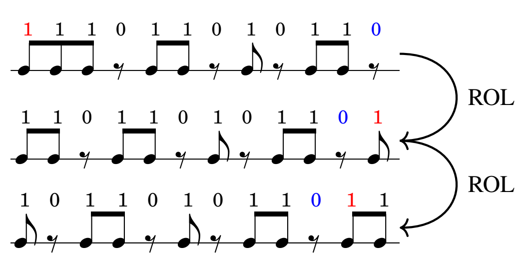

Figure 2: ROL operation applied to *Clapping music* pattern

Algorithm 1 Clapping music Algorithm

$P_1 \leftarrow [1,1,1,0,1,1,0,1,0,1,1,0]$
$P_2 \leftarrow P_1$
$Clap_1 \leftarrow repeat(P_1, 12*n-1)$
$Clap_2 \leftarrow repeat(P_2, n-1)$
for $i \leftarrow 1$ **until** $size(P_1)$ **do**
    $P_2 \leftarrow \texttt{ROL}(P_2, 1/8)$ ▷ 1/8 = eighth note
    $Clap_2 \leftarrow \texttt{concatenate}(Clap_2, \texttt{repeat}(P_2, n-1))$
end for
$Piece \leftarrow \texttt{overlay}(Clap_1, Clap_2)$

The *ROL* operation in Figure 2 is compositionally defined through segmentation and recombination. Let $(P_1, P_2) = \texttt{split}(P, \tau)$, where τ is a temporal split point. Then

$$\texttt{ROL}(P, \tau) = \texttt{concatenate}\big(\texttt{shift}(P_2, -\tau),\ P_1\big).$$

Tonal Hierarchy

The *Pattern* object operates over lists of events and can therefore be extended through specialized sub-languages. In our tonal music instantiation, we define sub-languages for rhythm, dynamics, and harmony/pitch-based events. These may interact to form higher-order musical structures composed within *Patterns*. From an MDL perspective, this factorizes the hypothesis space over distinct musical dimensions.

Here we focus on the pitch/harmony sub-language, grounded in the *tonal hierarchy*. The tonal hierarchy provides a cognitively grounded representation of pitch organization in tonal music (Krumhansl and Kessler, 1982), later refined for analyzing compositions in *Tonal Pitch Space* (Lerdahl, 2005).

In this framework, the tonal hierarchy models how listeners organize pitches into nested levels of structural importance (Figure 3). These levels reflect cognitive importance, and encode expectations about pitch attraction and melodic proximity. Common melodic patterns and abstractions (Schenker, 1935) could crucially be expressed using the tonal hierarchy as proposed by Tymoczko and others (Tymoczko, 2023). Each pitch event is associated with an instantiated hierarchical level specifying its role in the local harmonic context; all primitives below are evaluated in relation to their local harmony.

For pitch entities N, the primitive

$$\texttt{neighbor}(N, d, k) : N$$

returns the adjacent pitch of N in direction d at hierarchical level k.

Given two pitches N_1 and N_2, the interpolation operation

$$\texttt{passing}(N_1, N_2, k) : \mathrm{Seq}(N)$$

returns the intermediate pitches connecting N_1 to N_2 while respecting the constraints of level k.

Finally, sequential melodic trajectories are generated by

$$\texttt{get\_line}(N_1, n, d, k) : \mathrm{Seq}(N)$$

which returns a stepwise path of length n from N_1 in direction d at level k.

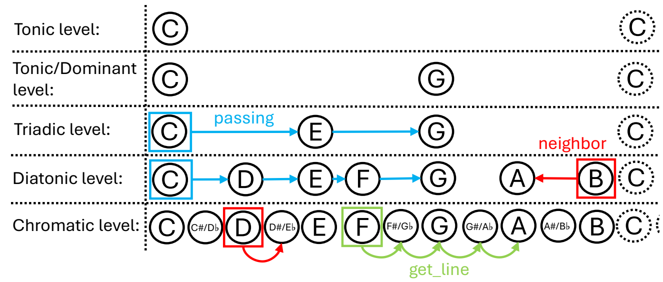

Figure 3: Tonal hierarchy of C Major

Together, these operations define a minimal set of domain-specific pitch primitives capturing both local transitions and structured melodic motion.

Figure 3 illustrates upper and lower chromatic neighbors (D → D# / E♭, B → A), passing structures between C and G yielding either arpeggiated [C,E,G] or diatonic [C,D,E,F,G] interpolations depending on level, and chromatic line generation from F ([F,F#,G,G#,A]). Chromatic nodes are shown with enharmonic pitch classes for visualization, while the implementation preserves pitch spelling distinctions (e.g., A# versus B♭) through a spelled-pitch representation (Finkensiep et al., 2022).

Polyphonic Example

Figure 4 shows the generation of the first bar of Bach's C Major prelude within our system. The prelude is based on broken arpeggios (Alfano, 2014; Drabkin, 1985; Lester, 1998). It begins on the tonic (C) and arpeggiates upward (`get_line(C,7,UP,Triadic)`). The final three notes may be interpreted either as an octave displacement of the preceding arpeggio, as shown here, or as a direct repetition of the previous three chord tones.

The rhythm is a uniform sixteenth-note texture without rests. The `add_rhythm` function is used as a convenience abstraction, since onsets and durations could alternatively be attached directly to pitches or generated through a rhythmic grammar (e.g. Rohrmeier, 2020). Polyphony arises from the first two notes, sustained as pedal tones over the full duration of the pattern. Finally, the pattern is repeated to fill the four beats of the first measure.

This compositional process can be encapsulated as a reusable function, allowing the pattern to be varied or recombined as needed, which is particularly useful for theme and variation as we will see in next section.

Results

Caprice 24 from 24 Caprices for Solo Violin, Op.1, Paganini

Figure 5 shows a simplified representation of the Paganini theme and its variations. The theme is in binary form: an 8-bar repeated i–V–i–V harmonic progression ending with a half-cadence, followed by a second 8-bar section subdivided into a descending fifths line and a cadential passage. A small set of functions are enough to generate the theme,

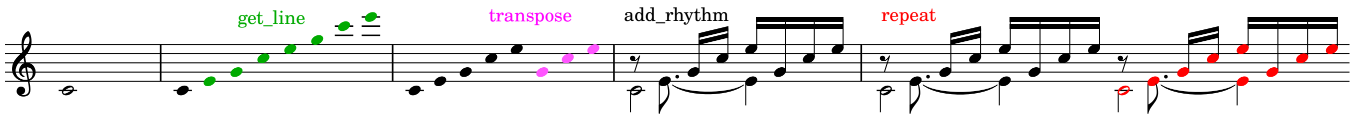

Figure 4: Generation of the first bar of Prelude I in C Major, Bach (BWV 846)

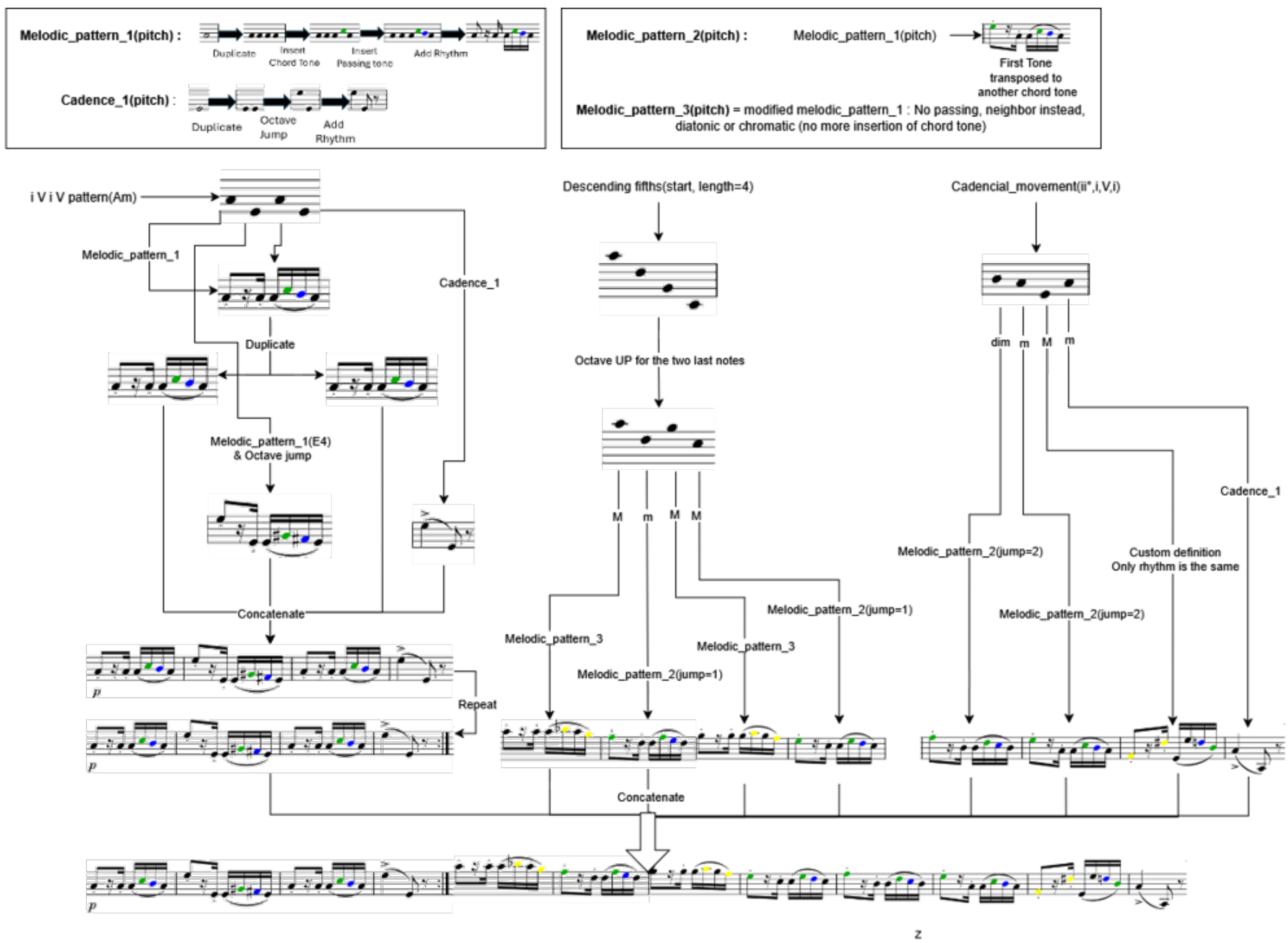

Figure 5: Main Theme Generation

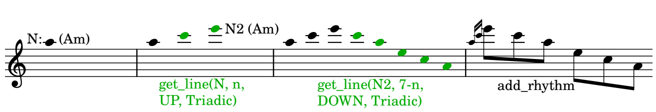

Figure 6: Generation of the pattern of Variation 1 (p_var1) with $N = A(Am)$ & $n = 2$

with melodic patterns in the second section systematically derived as variations of the first. This reuse creates a compact, self-referential structure: the theme maintains a consistent rhythmic texture across variations, while each variation introduces surface-level rhythmic differences without disrupting internal consistency.

Figure 7 presents a simplified symbolic representation of the first variation. As in the theme, the piece is constructed through the concatenation of bars, and the initial section is repeated verbatim. However, these structural properties are inherited directly from the theme and therefore do not need to be re-encoded, resulting in a more compact representation. At an abstract level, the variation preserves the same binary form and harmonic scaffold as the theme, with only a localized deviation in the antepenultimate bar, where a diminished chord replaces the augmented sixth. This exception can be integrated into the melodic schema of the theme itself, as it also supports theme and subsequent variations. As shown in Figure 6, melodically, the three principal contour types in the variation can be parameterized by a single value indicating the number of upward pitch steps from an anchor point, with the remaining motion proceeding downward. Aside from the E-major bar preceding the final cadence, which mirrors the exception already present in the theme, this parameterization accounts for the entire melodic surface of the variation.

Variation 2 is generated using the same harmonic structure

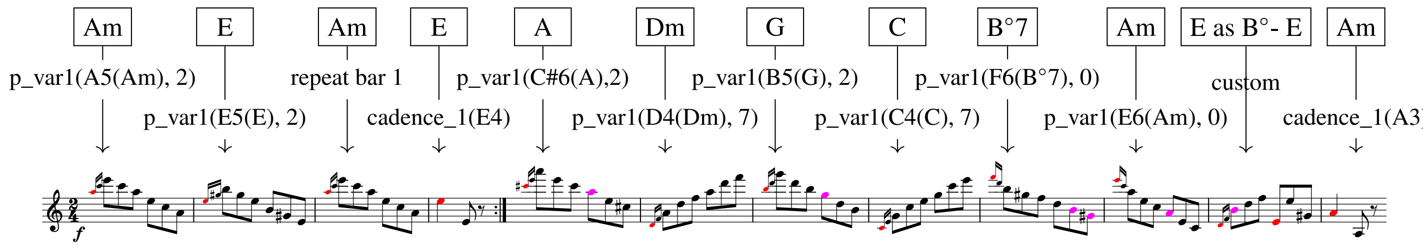

Figure 7: Variation 1 Generation with harmonic structure from theme.

Figure 8: First bar generation, Caprice 24, Var 2

and higher-level organization. As shown in Figure 8, the notes are generated by repeating an input note, then inserting chromatic neighbor tones, and adding an upward or downward line depending on the bar. The remaining bars are generated by a stepwise line directed toward the next harmonic anchor, optionally initiating from a lower chord tone in specific locations. These optional deviations occur in the second half of the piece and contribute to a stronger sense of closure. From a cognitive perspective, both variations illustrate how surface diversity can be generated through the systematic manipulation of a small number of symbolic parameters operating over a shared structural representation.

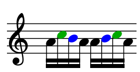

Figure 9: Melodic pattern symmetry

It is also worth noting that an additional half-bar pattern is inherited directly from the theme, as illustrated in Figure 9. When aligned on the same starting pitch, the two patterns are mirror-symmetric, corresponding to functional inversions of one another. As a result, the pattern can be abstracted independently of absolute pitch: starting from any anchor note, one can reuse the terminal segment of `melodic_generator_1` and apply a reversal operation. This illustrates how symmetry relations allow multiple surface realizations to be derived from a single transformation through minimal additional structure.

Compression

Compression in our framework operates at multiple levels. At the base, symbolic programs yield shorter descriptions than specifying each note and parameter independently. At a higher level, variations encoded conditionally on a theme require only the transformations and parameters that differ from the shared model. From an MDL perspective, this constrains the hypothesis space and binds free variables to established structure, substantially reducing conditional description length. This reuse of symbolic structure reflects a cognitively plausible strategy: new musical material emerges by adapting existing representations rather than constructing entirely new ones.

Following the Minimum Description Length (MDL) principle, description length is the sum of model cost and data cost given the model (P. Grünwald and Roos, 2019; P. D. Grünwald, 2007; Rissanen, 1978):

$$L(x) = L(M) + L(x \mid M), \tag{2}$$

where x is a musical object, M a generative model, and $L(\cdot)$ is measured in bits.

In a program-based representation, x is generated by symbolic operators O (e.g. transposition, neighboring motion) with parameter bindings Θ, giving

$$L_{\text{prog}} = L(O) + L(\Theta), \tag{3}$$

where O denotes the multi-set of operators used in the program and Θ the associated parameter bindings.

Conditional MDL encodes a variation v relative to a theme t, specifying only deviations from the shared model:

$$L(v \mid t) = L(\Theta_v) + L(\Delta_v), \tag{4}$$

with Θ_v the new parameters and Δ_v local structural exceptions. Compression is then quantified by

$$\text{Compression Ratio} = \frac{L_{\text{baseline}}(x)}{L_{\text{prog}}(x)}, \tag{5}$$

where $L_{\text{prog}}(x)$ denotes the program-based description length (knowing the theme when applicable). This ratio measures how many times more compact the structured program representation is compared to a non-compositional baseline encoding.

Following Sablé-Meyer et al., 2022, description length is approximated by counting symbolic program elements rather than probabilistic code lengths. Complexity is the number of primitive function calls plus free parameters, each assigned 1 unit cost. Defining a higher-level function incurs a one-time cost equal to the primitives used, with subsequent calls counting once, reflecting abstraction reuse. Free parameters only contribute if not fixed by the function definition; for example, once `melodic_pattern1` is defined, calls require only starting pitch and tonal hierarchy. This uniform-cost scheme provides a transparent, cognitively plausible MDL estimate. Program-based MDL values are reported relative to a fixed set of primitives and previously defined abstractions; only additional cost is counted.

As a baseline, each musical event is encoded independently, with three free parameters per note (onset, duration, pitch) or

2 for *Clapping music* (onset, duration), yielding a flat, non-compositional representation like MIDI. Expressive parameters are omitted for simplicity.

Piece	Baseline (units)	Prog. MDL (units)	Cond. MDL (units)	Comp. Ratio
Theme	252	87	–	2.90
Variation 1	330	80	67	4.93
Variation 2	411	81	69	5.96
Clapping Music	4992	175	–	28.53
1st Bar Prelude	48	17	–	2.82

Table 1: Program-based and conditional description lengths measured in uniform symbolic units. The baseline column indicates the reference model relative to which conditional MDL is computed.

Table 1 summarizes description lengths for the theme and first two variations. Symbolically, variations have equal or greater length than the theme due to increased surface activity. With program-based conditional encoding, however, description length drops sharply, reflecting reuse of higher-level structures like harmonic scaffolding, formal organization, rhythmic schemas, and melodic operators. The theme thus functions as a shared model: once encoded, variations require fewer additional parameters and transformations. Compression ratios confirm that Paganini's variations achieve surface diversity through systematic manipulation of a compact symbolic core rather than new material.

Discussion

The present framework is a proof of concept showing the representations and patterns involved in tonal composition rather than providing a comprehensive model of musical behavior. Its primary contribution is representational correctness: it formalizes symbolic primitives, hierarchical organizations, and compositional operations central to tonal music pedagogy and analysis, which musicians explicitly manipulate when composing, learning, and varying material (Gjerdingen, 2007; Jackendoff, 2002; Temperley, 2006). It specifies a cognitively plausible representational level without making claims about neural or procedural implementation (Glennerster, 2007; Marr, 2010).

From a symbolic cognition and Language of Thought (LoT) perspective, the framework operationalizes musical structure as reusable symbolic expressions combined via a finite set of rules (Fodor, 1975). Patterns, tonal hierarchies, and transformations function like expressions and operators in a domain-specific symbolic language, supporting systematicity and productivity. The claim is not that musical cognition reduces to rule execution, but that tonal music provides a culturally stabilized domain in which symbolic structures are externalized transparently. Musical scores thus offer a clear window onto symbolic cognition, encoding hierarchical relations and transformations in inspectable artifacts (Cohn, 2013).

Using Minimum Description Length (MDL) (Chater and Vitányi, 2003; Rissanen, 1978) clarifies the cognitive relevance of these representations. Musical variations admit shorter symbolic descriptions when a theme is present, suggesting tonal abstractions act as compression mechanisms that facilitate memory, generalization, and creative reuse by minimizing descriptive complexity, rather than merely enforcing stylistic convention.

Beyond descriptive adequacy, the framework generates empirically testable predictions. Materials with shorter symbolic descriptions should be learned and recalled more efficiently (Alfano, 2014; Gruhn, 2002; StGeorge et al., 2012). Judgments of thematic similarity should correlate with symbolic distances over patterns and hierarchies. In production, trained musicians should preferentially generate variations reusing existing symbolic structures, reflecting implicit biases toward abstraction and reuse (Simonton, 1999).

Limitations include the hand-specified nature of abstractions and the focus on representational adequacy rather than modeling acquisition or cognitive processes. Future work could integrate program induction and similarity metrics to align more closely with learning-based and experimental approaches (Lake et al., 2017). Automatic pattern discovery could model abstraction learning by identifying recurring structures (Meredith et al., 2002; Ren et al., 2025), and library learning methods could infer reusable symbolic components supporting compositionality and description-length reduction (Bowers et al., 2023; Dechter et al., 2013; Ellis et al., 2018; Jiang et al., 2024).

Conclusion

This paper presented a proof-of-concept framework modeling tonal music as an executable symbolic system grounded in theories of symbolic cognition and the LoT. By representing musical material as structured patterns and compositional operations as symbolic transformations, tonal composition is formalized as the manipulation of discrete symbolic structures akin to cognitive programs.

Analyses of Clapping Music and Paganini's Caprice No. 24 show that musically meaningful abstractions yield shorter symbolic descriptions than note-level encodings. From a MDL perspective, this demonstrates that tonal music relies on compact, compositional representations consistent with principles of systematicity, productivity, and parsimony. The framework is not a full model of musical cognition, but an explicit instantiation of the symbolic structures underlying tonal composition.

More broadly, it illustrates how musical scores function as externalized cognitive programs, linking music-theoretical categories to executable symbolic representations. This provides a bridge between music theory, computational modeling, and cognitive theories of representation, and establishes a foundation for future empirical and theoretical studies of symbolic musical cognition.

Acknowledgments

We thank all members of the Digital and Cognitive Musicology Lab at EPFL for their valuable feedback. In particular, we thank Zeng Ren for insightful comments and valuable suggestions throughout the development of this project. This research was supported by the Swiss National Science Foundation under grants Nos. 10000183 and 215701. The authors thank Mr. Claude Latour for generously supporting this research through the Latour chair in digital musicology.

References

Alfano, D. (2014). Learning Bach's Short Preludes: A Sequential Approach. *Clavier Companion*.

Bowers, M., Olausson, T. X., Wong, L., Grand, G., Tenenbaum, J. B., Ellis, K., & Solar-Lezama, A. (2023). Top-Down Synthesis for Library Learning. *Proceedings of the ACM on Programming Languages*, *7*(POPL), 1182–1213. https://doi.org/10.1145/3571234

Chaitin, G. J. (1966). On the Length of Programs for Computing Finite Binary Sequences. *J. ACM*, *13*(4), 547–569. https://doi.org/10.1145/321356.321363

Chater, N., & Vitányi, P. (2003). Simplicity: A unifying principle in cognitive science? *Trends in Cognitive Sciences*, *7*(1), 19–22. https://doi.org/10.1016/s1364-6613(02)00005-0

Cohn, N. (2013, December). *The Visual Language of Comics: Introduction to the structure and cognition of sequential images*. Bloomsbury.

Cohn, N. (2020). Your Brain on Comics: A Cognitive Model of Visual Narrative Comprehension. *Topics in Cognitive Science*, *12*(1), 352–386. https://doi.org/10.1111/tops.12421

Dechter, E., Malmaud, J., Adams, R. P., & Tenenbaum, J. B. (2013). Bootstrap learning via modular concept discovery. *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, 1302–1309.

Deutsch, D. (1999). *The psychology of music, 2nd ed.* Academic Press.

Drabkin, W. (1985). A Lesson in Analysis from Heinrich Schenker: The C Major Prelude from Bach's Well-Tempered Clavier, Book I. *Music Analysis*, *4*(3), 241–258. https://doi.org/10.2307/854097

Ellis, K., Morales, L., Sablé-Meyer, M., Solar-Lezama, A., & Tenenbaum, J. (2018). Learning Libraries of Subroutines for Neurally– Guided Bayesian Program Induction. *Advances in Neural Information Processing Systems*, *31*.

Finkensiep, C., Lieck, R., & Rohrmeier, M. (2022). A Family of Libraries for Working With Pitches and Intervals. *Ismir 2022 Hybrid Conference*.

Fodor, J. (1975). *The Language of Thought*. Harvard University Press.

Fodor, J., & McLaughlin, B. P. (1990). Connectionism and the problem of systematicity: Why Smolensky's solution doesn't work. *Cognition*, *35*(2), 183–204. https://doi.org/10.1016/0010-0277(90)90014-B

Gjerdingen, R. (2007). *Music in the Galant Style*. Oxford University Press.

Glennerster, A. (2007). Marr's vision: Twenty-five years on. *Current Biology*, *17*(11), R397–R399. https://doi.org/10.1016/j.cub.2007.03.035

Gruhn, W. (2002). Phases and Stages in Early Music Learning. A longitudinal study on the development of young children's musical potential. *Music Education Research*, *4*(1), 51–71. https://doi.org/10.1080/14613800220119778

Grünwald, P., & Roos, T. (2019). Minimum Description Length Revisited. *International Journal of Mathematics for Industry*, *11*(01), 1930001. https://doi.org/10.1142/S2661335219300018

Grünwald, P. D. (2007, March). *The Minimum Description Length Principle* (F. Bach, Ed.). MIT Press.

Haack, J. K. (1991). Clapping Music—a Combinatorial Problem. *The College Mathematics Journal*, *22*(3), 224–227. https://doi.org/10.1080/07468342.1991.11973385

Haack, J. K. (1998). The Mathematics of Steve Reich's "Clapping Music". *Bridges: Mathematical Connections in Art, Music, and Science*, 87–92.

Hörnel, D., & Menzel, W. (1998). Learning Musical Structure and Style with Neural Networks. *Computer Music Journal*, *22*(4), 44–62. https://doi.org/10.2307/3680893

Huron, D. (2008, January). *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press.

Jackendoff, R. (2002, January). *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780198270126.001.0001

Jiang, G., Hofer, M., Mao, J., Wong, L., Tenenbaum, J., & Levy, R. (2024). Finding structure in logographic writing with library learning. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *46*(0).

Johnson, D. D., Keller, R. M., & Weintraut, N. (2017). Learning to create jazz melodies using a product of experts. *ICCC*, 151–158.

Kolmogorov, A. (1965). Three approaches to the definition of the concept "quantity of information". *Problemy Peredachi Informatsii*, 3–11.

Krumhansl, C. L., & Kessler, E. J. (1982). Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review*, *89*(4), 334–368. https://doi.org/10.1037/0033-295X.89.4.334

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *The Behavioral and Brain Sciences*, *40*, e253. https://doi.org/10.1017/S0140525X16001837

Lerdahl, F. (2005, January). *Tonal Pitch Space*. Oxford University Press.

Lerdahl, F., & Jackendoff, R. S. (1985, May). *A Generative Theory of Tonal Music*. MIT Press.

Lester, J. (1998). J. S. Bach Teaches Us How to Compose: Four Pattern Preludes of the "Well-Tempered Clavier". *College Music Symposium*, *38*, 33–46.

Li, M., & Vitányi, P. (2008). *An Introduction to Kolmogorov Complexity and Its Applications*. Springer. https://doi.org/10.1007/978-0-387-49820-1

Marr, D. (2010, July). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. MIT Press.

Meredith, D., Lemstrom, K., & Wiggins, G. A. (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, *31*(4), 321–345. https://doi.org/10.1076/jnmr.31.4.321.14162

Newell, A. (1994). *Unified Theories of Cognition*. Harvard University Press.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Prentice-Hall.

Nishimura, S., & Marui, A. (2025). Pytakt: A Python library for symbolic music description, generation, and real-time processing. *Journal of New Music Research*, *54*(1), 71–87. https://doi.org/10.1080/09298215.2025.2540434

Pylyshyn, Z. W. (1984, June). *Computation and Cognition: Toward a Foundation for Cognitive Science*. The MIT Press. https://doi.org/10.7551/mitpress/2004.001.0001

Reich, S. (2002, April). *Writings on Music, 1965-2000*. Oxford University Press.

Ren, Z., Guan, X., & Rohrmeier, M. (2025, April). A Computational Cognitive Model for Processing Repetitions of Hierarchical Relations. https://doi.org/10.48550/arXiv.2504.10065

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*(5), 465–471. https://doi.org/10.1016/0005-1098(78)90005-5

Rohrmeier, M. (2020). Towards a formalization of musical rhythm. *Proceedings of the 21st International Society for Music Information Retrieval Conference*. https://doi.org/10.5281/zenodo.4245508

Sablé-Meyer, M., Ellis, K., Tenenbaum, J., & Dehaene, S. (2022). A language of thought for the mental representation of geometric shapes. *Cognitive Psychology*, *139*, 101527. https://doi.org/10.1016/j.cogpsych.2022.101527

Schenker, H. (1935). *Free Composition*. Simon & Schuster Books For Young Readers.

Sethares, W. A., & Toussaint, G. T. (2015). Expressive Timbre and Timing in Rhythmic Performance: Analysis of Steve Reich's Clapping Music. *Journal of New Music Research*, *44*(1), 11–24. https://doi.org/10.1080/09298215.2014.935736

Simonton, D. K. (1999). *Origins of genius: Darwinian perspectives on creativity*. Oxford University Press.

Solomonoff, R. J. (1964). A formal theory of inductive inference. Part I. *Information and Control*, *7*(1), 1–22. https://doi.org/10.1016/S0019-9958(64)90223-2

StGeorge, J. M., Holbrook, A. P., & Cantwell, R. H. (2012). Learning patterns in music practice: Links between disposition, practice strategies and outcomes. *Music Education Research*, *14*(2), 243–263. https://doi.org/10.1080/14613808.2012.685454

Taube, H. (1991). Common Music: A Music Composition Language in Common Lisp and CLOS. *Computer Music Journal*, *15*(2), 21–32. https://doi.org/10.2307/3680913

Temperley, D. (2004). *The Cognition of Basic Musical Structures*. MIT Press.

Temperley, D. (2006). *Music and Probability*. The MIT Press. https://doi.org/10.7551/mitpress/4807.001.0001

Tymoczko, D. (2006). The Geometry of Musical Chords. *Science*, *313*(5783), 72–74. https://doi.org/10.1126/science.1126287

Tymoczko, D. (2023). *Tonality: An Owner's Manual*. Oxford University Press.

Tymoczko, D. (2024). Refactoring Musical Thought (Keynote). *Proceedings of the 12th ACM SIGPLAN International Workshop on Functional Art, Music, Modelling, and Design*, 1. https://doi.org/10.1145/3677996.3689597