applied sciences

MDPI

Article

SymCHM—An Unsupervised Approach for Pattern Discovery in Symbolic Music with a Compositional Hierarchical Model

Matevž Pesek [1,*], Aleš Leonardis [2] and Matija Marolt [1]

[1] Faculty of Computer and Information Science, University of Ljubljana, Ljubljana 1000, Slovenia; matija.marolt@fri.uni-lj.si

[2] School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK; a.leonardis@cs.bham.ac.uk

* Correspondence: matevz.pesek@fri.uni-lj.si; Tel.: +386-1-4798-259

Academic Editor: Meinard Müller
Received: 13 September 2017; Accepted: 1 November 2017; Published: 4 November 2017

Abstract: This paper presents a compositional hierarchical model for pattern discovery in symbolic music. The model can be regarded as a deep architecture with a transparent structure. It can learn a set of repeated patterns within individual works or larger corpora in an unsupervised manner, relying on statistics of pattern occurrences, and robustly infer the learned patterns in new, unknown works. A learned model contains representations of patterns on different layers, from the simple short structures on lower layers to the longer and more complex music structures on higher layers. A pattern selection procedure can be used to extract the most frequent patterns from the model. We evaluate the model on the publicly available JKU Patterns Datasetsand compare the results to other approaches.

Keywords: music information retrieval; compositional modelling; pattern discovery; symbolic music representations

1. Introduction

In music, hierarchical representations are intuitive when one considers its spectral and temporal structures. In an analytical sense, the Generative Theory of Tonal Music (GTTM) by Lerdahl and Jackendoff [1] offers an approach of explicit hierarchical music modelling in musicology, well known in contemporary music theory. Although GTTM mostly relies on expert rules, the concept of hierarchical structuring seems reasonable, derived from the humans' search for structure in consciously perceived surroundings. There are several attempts to build a system capable of automatic analysis supported by the GTTM and Schenkerian analysis [2–4]. Several other rule-based models were also researched in Music Information Retrieval (MIR) and related fields [5,6]. Furthermore, the hierarchical models abound in analysis of music perception from the point of view of computational biology and neuroscience [7,8].

In parallel to explicit hierarchical representations, a variety of new approaches emerged under a common name of deep learning [9]. Several neural-network-based approaches have been proposed for melody transcription (e.g., [10]), genre classification (e.g., [11]), onset detection (e.g., [12]), drum pattern analysis (e.g., [13]) and chord estimation (e.g., [14]). The idea behind a deep learning algorithm is to construct multiple levels of data abstraction: a hierarchy of features. The high-level representations in the training data are reflected in the hierarchy. However, the encoded knowledge is implicit and is difficult to explain in a transparent (non black-box) way. Therefore, although deep learning enables unsupervised learning of features and achieves good results on a variety of tasks, it is not very appropriate for pattern discovery in music where explicit explanations of input are desired.

The discovery of repeated patterns is a known problem in different domains, including computer vision (e.g., [15]), bioinformatics (e.g., [16]) and music information retrieval (MIR). Although a common problem, its definition, as well as pattern discovery algorithms, significantly differs across these fields. In music, the importance of repetition has been addressed and discussed by a number of music theorists (e.g., [17]) and, more recently, also by researchers who develop algorithms for semi-automatic music analysis, such as one described by Marsden [4]. In the MIR field, an initiative for a common definition of different tasks was formalized into the Music Information Retrieval Evaluation eXchange (MIREX), in an attempt to compare different approaches. MIREX is a community-based framework for formal evaluation of algorithms and techniques related to MIR [18]. The MIREX community established several tasks dealing with patterns and structures in music, including structural segmentation, symbolic melodic similarity and pattern matching, and pattern discovery.

The aim of the discovery of repeated themes and sections task is to find repetitions which represent one of the more significant aspects of a music piece [19]. The MIREX task definition states "the algorithms take a piece of music as input, and output a list of patterns repeated within that piece" [20]. The task may also seem similar to the well-known pattern matching task [21], However, while a pattern matching algorithm aims to find the place of a searched pattern within a dataset and usually has a clear quantitative relation between a query and a match, a discovery of repeated patterns finds locations of multiple similar sequences of data in the dataset, without any information about the searched pattern. The definition of a pattern has been troubling researchers since the beginning; while a pattern may come as an intuitive representation with a repetitive substance, patterns in music are more difficult to define and are usually formalized using theoretical rules, specific to the music era and genre. In the discovery of repeated themes and sections task, a pattern is defined as "a set of on-time-pitch pairs that occurs at least twice (i.e., is repeated at least once) in a piece of music. The second, third, etc. occurrences of the pattern will likely be shifted in time and perhaps also transposed, relative to the first occurrence." [20]. As noted by Wang et al. [22], the pattern discovery task differs from the structural segmentation task, where segments cover the whole music piece and represent disjoint sets of events. In the pattern discovery task, patterns may partially overlap or be subsets of another pattern. However, some of the approaches mentioned in this section (e.g., [23,24]) perform pattern discovery by calculating a set of non-overlapping patterns.

A variety of approaches has been proposed for pattern discovery in music in the past years. Conklin and Anagnostopoulou [25] proposed a multiple viewpoint pattern discovery algorithm based on a suffix-tree. For a selected viewpoint (a transformation of a musical event into an abstract feature) the algorithm builds a suffix tree of viewpoint sequences (transformed music pieces). After selecting patterns which meet specified frequency and significance thresholds, the leafs of the suffix tree are reported as longest significant patterns in the corpus. Conklin and Bergeron [24] apply two algorithms, using viewpoints which represent abstract properties of musical notes for statistical modelling of melody [26]. A viewpoint is thus a function that computes values for events in a sequence; a pattern is a sequence of such feature sets, where the latter represent a logical conjunction of multiple viewpoints. The authors present a complete algorithm which can find all 'maximal frequent patterns' and an optimization algorithm using a faster heuristic approach, where the found patterns may not always be the maximal frequent patterns. The maximal frequent pattern represents a pattern whose component feature set cannot be further specialized without the pattern becoming infrequent. Rolland [27] presents the FlExPat (Flexible Extraction of Patterns) algorithm for extracting sequential patterns from sequences of data. The algorithm first identifies equipollent passage pairs and produces a similarity graph, representing the relations between each two passages; patterns are extracted from the similarity graph. The author evaluated the approach on a set of ten Charlie Parker solos from the subset of Owens' corpus [28] and reported a satisfactory pattern extraction of a large number of the annotated patterns. Cambouropoulos et al. [23] introduced an approach for extraction of patterns from abstract strings of symbols, allowing for a partial overlap of various abstract symbolic classes. They also focused on time complexity of their solution and addressed the problem of approximate pattern matching. Based on their previous work [29], they presented the PAT algorithm for segmentation based on maximal repeated patterns. Besides discovering the patterns, and subject and counter-subject entries in fugues, Meredith [30] described multiple point-set compression algorithms, including several COSIATEC and COSIATECCompress approaches and Forth's algorithm. The author evaluated these approaches on three music analysis tasks: the classification of folk song melodies into tune families, discovering entries of subjects and counter-subjects in fugues, and the discovery of repeated themes and sections in polyphonic works task. Meredith [31] also evaluated his SIATECCompressSegment algorithm for the task, which is a greedy compression algorithm based on the previously introduced SIATEC approach [19]. The algorithm evaluates patterns based on assumption that perceptually interesting patterns correspond to Maximal Translatable Patterns (MTP). The approach produces a compact encoding of a musical piece, defined by a point-set representation, in form of a set of Translational Equivalence Classes (TEC) of MTPs. The MTP with a defined particular vector is a set of points, which can be translated by that vector to give other points in the point-set representation. The authors observed that the MTPs often correspond to perceptually significant repeated patterns in music. The TEC defines a set of all patterns which are translationally equivalent to a pattern defining the specific TEC. The SIATECCompressSegment approach generates an ordered list of TECs which may overlap (in contrast to other related versions such as COSIATEC).

Recently, Velarde and Meredith [32] extended a previously introduced approach to melodic segmentation [33] for melodic classification and segmentation, where the symbolic input is first segmented, then compared and hierarchically clustered. Finally, the clusters are ranked, taking into account the cumulative length of all occurrences within each cluster. Based on their results, it can be assumed that the output is additionally filtered by a threshold defining the number of output patterns. Lartillot [34] introduced the PatMinr algorithm [35] which uses an incremental one-pass approach to identify pattern occurrences. To avoid redundancy, the author addresses two issues: closed pattern mining, which filters out the patterns that have more occurrences than their more specific patterns, thus providing more robust patterns, and pattern cyclicity, which removes redundant matches for successive occurrences of a single underlying pattern. The most recent approach submitted to the MIREX task by Ren [36] also employs a closed pattern approach commonly used in data mining. Nieto and FarBood [37] proposed the MotivesExtractor which obtains a harmonic representation of the audio or symbolic input and extracts patterns based on a produced self-similarity matrix. Using a score-based greedy algorithm ([38]) the approach extracts repeated segments, allowing the patterns to overlap. Finally, the segments are grouped into clusters and provided in the algorithm's output as patterns.

In contrast to the existing hierarchical and deep approaches, the Compositional Hierarchical Model (CHM) presented in this paper is a transparent deep architecture. The model provides an explicit (transparent) encoding of concepts, learned in an unsupervised manner, thus merging the benefits of explicit and deep hierarchical models in MIR. The CHM is built around the premise that the repetitive nature of patterns can be captured by observing statistics of occurrences of their sub-patterns, thus providing a hierarchy of the analysed symbolic music representation(s) [39]. Similar to other approaches that build a tree of patterns based on their subsumption (e.g., [25]), the CHM first extracts small atomic patterns and builds complex patterns as compositions of these atomic patterns. Its ability to concurrently provide multiple pattern hypotheses on several levels of complexity and their transparent descriptions makes it very suitable for pattern extraction, as patterns may overlap or be mutually included.

The compositional hierarchical model was first introduced by Pesek et al. [40] and was evaluated for several MIR tasks, including automated chord estimation and multiple fundamental frequency estimation [41]. In the paper, we present an adaptation of the model for analysis of Symbolic music (SymCHM) applied to the task of finding repeated patterns and sections. Instead of finding compositions in a frequency-magnitude audio representation, the adjusted model searches for compositions of symbolic events in the time-pitch-onset domain. The model learns a hierarchy of patterns; the transparent nature of the model allows the user to explore and analyse a music piece by observing the hierarchy of pattern occurrences. For the automatic discovery of repeated patterns, the patterns represented in the hierarchy are extracted. We analyse the model output and propose an extension of the model named SymCHMMerge, which refines the extracted patterns.

The contributions of this paper are as follows: the compositional hierarchical model for symbolic music analysis that can learn hierarchical melodic structures in an unsupervised manner is presented. An application of the model to the task of finding repeated patterns and sections is evaluated. The improved pattern extraction and merging approach from knowledge encoded in the model (SymCHMMerge) is proposed and analysed.

The paper is structured as follows: we present the SymCHM in Section 2, describe its application and extension to pattern extraction in Section 3 and present its evaluation and error analysis in Section 4. We conclude the paper with an overview of other possible applications of the presented model and outline future work in Section 5.

2. The Symbolic Compositional Hierarchical Model

The Symbolic Compositional Hierarchical Model (SymCHM) is derived from the CHM [40,41], which in turn was inspired by an approach for object categorization in computer vision, named the learned Hierarchy of Parts (lHoP) [42]. The SymCHM provides a hierarchical representation of a symbolic music piece, from individual notes on the lowest layer, up to complex musical patterns on higher layers. It is based on a hierarchical decomposition of music into atomic blocks, denoted as parts (not to be confused with 'voice' or 'vocal/instrumental part'. This denomination is used to retain the consistency in relation to the lHoP). According to their musical complexity, parts are structured across several layers, whereby parts on higher layers form compositions of parts on lower layers. A part can therefore describe a simple individual event as well as a complex composition of events. While events in the original compositional hierarchical model represent spectral audio features (frequencies, pitch partials and pitches), the SymCHM models notes and their compositions into melodic patterns.

2.1. Model Description

2.1.1. Compositional Layers

The SymCHM consists of an input layer $\mathcal{L}_0$ and several compositional layers $\{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$. Each compositional layer $\mathcal{L}_n$ contains a set of parts $\{P_1^n, \ldots, P_{M_n}^n\}$, which are formed as compositions of parts from the previous layer $\mathcal{L}_{n-1}$. The parts on the layer $\mathcal{L}_{n-1}$ may form any number of compositions on the layer $\mathcal{L}_n$, which enables their effective reuse and thus learning of compact models, as shown later in this paper. A hierarchy of parts is illustrated in Figure 1.

The SymCHM retains part definitions of the original CHM model. The i-th composition on the layer $\mathcal{L}_n$, denoted P_i^n, is defined as:

$$P_i^n = \{P_{k_0}^{n-1}, \{P_{k_j}^{n-1}, (\mu_j, \sigma_j)\}_{j=1}^{K-1}\}. \tag{1}$$

P_i^n is a composition of K parts from the layer $\mathcal{L}_{n-1}$, called subparts. The composition is governed by parameters $\mu_{1,\ldots,K-1}$ and $\sigma_{1,\ldots,K-1}$, which model relationships between the subparts. In contrast to most existing hierarchical and deep approaches, the CHM encodes compositions in a relative rather than absolute manner. This is achieved by encoding the relative distance (offset) between each subpart $P_{k_j}^{n-1}$, from the first subpart $P_{k_0}^{n-1}$, called the central part. The offset is encoded as a Gaussian with parameters μ_j and σ_j. In SymCHM, offsets are modelled in semitones in the pitch domain (a semitone is the smallest musical interval commonly used in Western tonal music), thus a composition encodes the semitone distance between patterns represented by various subparts. Currently, the standard deviation σ_j is set to a small fixed value, which does not allow for deviations from the offset encoded by μ_j. In future work we may relax this condition to potentially achieve similar robustness as in chromatic to morphetic pitch translation [43]. As an example, the part P_2^3 in Figure 1 represents a composition of two subparts with offset 2 ($\mu = 2$), meaning its pattern is a concatenation of two sub-patterns spaced two semitones apart. All compositions and their parameters (μ, σ) are learned in an unsupervised manner as explained in Section 2.2.

Such relative encoding of knowledge enables the model to learn position-independent concepts, which in turn enables learning of compact models from small datasets, which still generalize well [41]. This is an advantage over most neural network deep approaches, which encode concepts in an absolute manner and therefore need very large datasets to train properly.

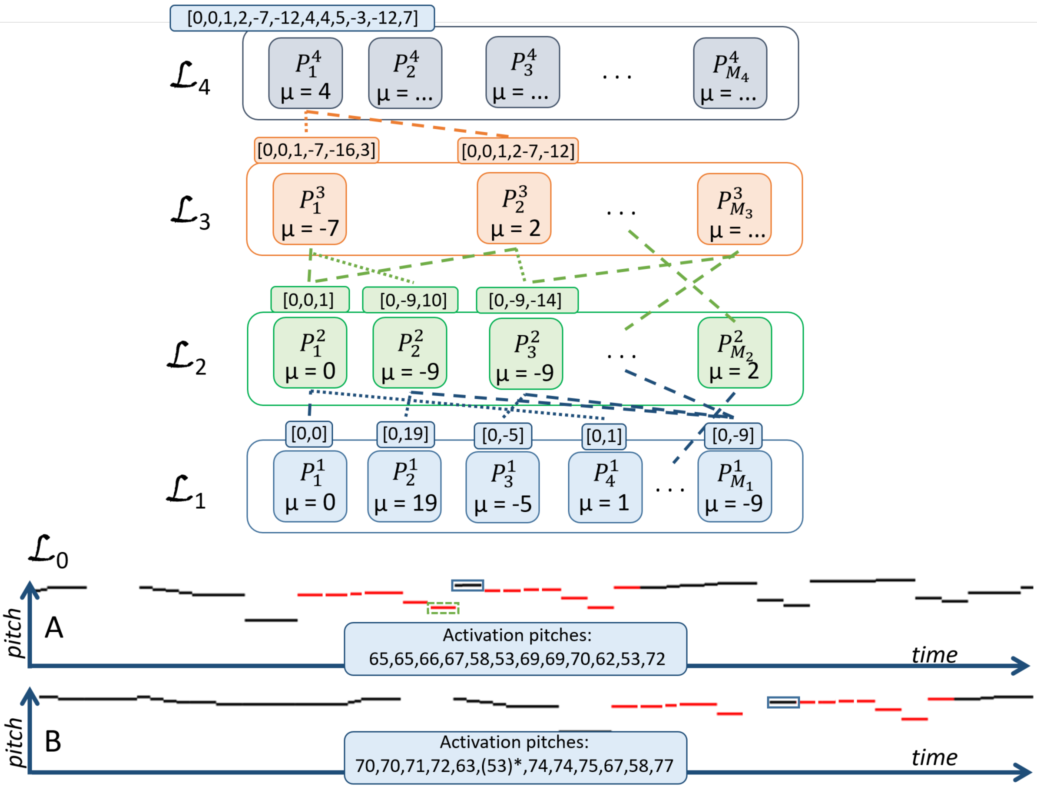

Figure 1. The symbolic compositional hierarchical model. The input layer corresponds to a symbolic music representation (a sequence of pitches). Parts on higher layers are compositions of lower-layer parts (depicted as connections between parts, the parameter μ is given in semitones). The structure of a part is displayed above each part in the figure, represented by a sequence of pitch values relative to the first subpart (e.g., [0,0,1] for the part P_1^2). A part may be contained in several compositions, e.g., $P_{M_1}^1$ is a part of compositions P_2^2 and P_3^2. The entire structure is transparent, thus we can observe the entire sub-tree of the part P_1^4. A part activates, when (a part of) the pattern it represents is found in the input. As an example, P_1^4 activates twice (Inputs A and B), however there are differences in the found patterns. Pattern A is positioned five semitones higher than B; Pattern B is missing one event (dotted green rectangle); and the pitch of one event (blue rectangle) differs between the two patterns.

2.1.2. Activations: Occurrences of Patterns

An activation of a part corresponds to the presence of the concept it encodes (melodic pattern in SymCHM) in the model input. An activation has three components: location and onset time, which map the relative pattern representation onto a specific MIDI (Musical Instrument Digital Interface technical standard) pitch and a time position within the input sequence of events (thus making it absolute) and magnitude, representing its strength.

A part will activate at a given location if all of its subparts are activated with magnitude greater than zero (this condition is relaxed with hallucination, which we introduce later in this section). A part can concurrently activate at different locations and times, which indicates multiple occurrences of its concept in the input representation. In terms of the repeated pattern discovery task, each activation of a part can be observed as a pattern occurrence: a repetition of the pattern encoded by the observed part.

More formally, the activation A is defined as a triplet $\langle A_L, A_T, A_M \rangle$ of location, time and magnitude. The activation location A_L and the time A_T of the part P_i^n are defined as:

$$\begin{aligned} A_L(P_i^n) &= A_L(P_{k_0}^{n-1}) \\ A_T(P_i^n) &= A_T(P_{k_0}^{n-1}). \end{aligned} \tag{2}$$

The compositions therefore propagate their locations and onset times upwards through the hierarchy. Such propagation can be usefully employed as an indexing mechanism and allows for a top-down analysis of activations.

The activation magnitude represents the strength of the composition's match with the input and is defined as a weighted sum of subpart magnitudes:

$$A_M(P_i^n) = \tanh\left(\tfrac{1}{K}\textstyle\sum_{j=0}^{K-1} w_j A_M(P_{k_j}^{n-1})\right), \tag{3}$$

where the weights w_j are defined by the match between the learned and the observed relative subpart pitch locations and bounded by the difference in their activation times:

$$\begin{aligned} w_j &= \begin{cases} 1: & j = 0 \\ \mathcal{N}(\delta_{Lj}, \mu_j, \sigma_j): & j > 0 \wedge \delta_{Tj} < \tau_W \\ 0: & \delta_{Tj} \geq \tau_W \end{cases} . \\ \delta_{Lj} &= A_L(P_{k_j}^{n-1}) - A_L(P_{k_0}^{n-1}) \\ \delta_{Tj} &= A_T(P_{k_j}^{n-1}) - A_T(P_{k_0}^{n-1}) \end{aligned} \tag{4}$$

The motivation behind the usage of *tanh* function introduced in Equation (3) is retained from neural-network-based architectures: it provides a saturated output with the maximum limited to one. Any other function could be used to calculate the magnitude of the activation, but the hyperbolic tangent function possesses several interesting properties: it is a monotonically increasing function with a smooth gradient and has a value close to one as it approaches infinity. Since the activation magnitudes are directly used to calculate activations on a higher layer, the output of the function needs to be normalized.

The parameter τ_W represents the maximal difference between activation times of two subparts (time distance of two patterns) which still produces an activation. Such a limit must be imposed in order to avoid a combinatorial explosion of possible compositions. If subpart activations fall within this time window, their activation magnitude is calculated according to the match between their observed (δ_{Lj}) and their learned (μ_j, σ_j) relative pitch distances. A part will activate with maximal magnitude when its subparts activate at pitch distances according to the learned representation encoded by μ_j and σ_j. Note that onset times do not directly influence the activation magnitude. Thus, the activation strength of a pattern is not dependent on the temporal distance between its sub-patterns (within τ_W) and remains the same whether they are adjacent or separated by other events, allowing for gaps between sub-patterns.

2.1.3. The Input Representation and Input Layer

A symbolic music representation encoding note pitches and onset times represents input to the SymCHM. Any symbolic encoding that includes these values can be used, such as MusicXML, MIDI or text-based representations; the latter two are also available for the MIREX pattern discovery task.

We can thus define the input representation as a set of note onset (e.g., in seconds) and note pitch (e.g., MIDI pitch) tuples $\mathcal{S} = \{(N_o, N_p)\}$.

The input layer of SymCHM $\mathcal{L}_0$ models such a symbolic music representation. It consists of a single atomic part P_1^0, which activates for all note events as:

$$A = \langle A_L(P_1^0), A_T(P_1^0), A_M(P_1^0)\rangle \leftarrow \langle N_p, N_o, 1\rangle \tag{5}$$

Thus, the activation locations A_L are equal to note pitches, the onset times A_T to note onsets, while the magnitude A_M is assumed to be 1 for all events (it can also represent note dynamics, if greater importance is to be put on accented notes).

An example of a learned hierarchy is shown in Figure 1. The part P_1^0 is activated for each input note event. The parts on the first layer represent intervals, e.g., P_4^1 represents a minor second (offset one semitone) and is activated for all such intervals in the input regardless of gaps, with notes spaced maximally τ_W apart. P_1^4 represents a sequence of note events defined by a series of offsets [0,0,1,2,−7,−12,4,4,5,−3,−12,7] and is activated at MIDI locations 65 and 70.

2.2. Constructing a Hierarchy of Parts

The model is built layer-by-layer with unsupervised learning on a single or multiple musical pieces. In the 'intra-opus' pattern discovery task experiment described in this paper, we build a model for each musical piece separately.

The learning process is an optimization problem, where for each layer a set of all possible part compositions of the layer is searched for a minimal subset of compositions that covers a maximal amount of events in the training set. The learning process is driven by statistics of part activations that capture regularities in the input data. It consists of two main steps: (1) finding a set of all possible compositions, denoted candidate compositions, and (2) selecting compositions that explain a maximal amount of events in the training set.

To construct a new layer $\mathcal{L}_n$, a set of new candidate compositions $\mathcal{C}$, which will be considered for inclusion in the new layer, is first formed (Step 1). This set of candidate compositions is obtained by inferring the hierarchy with the training data and generating activations of parts layer-by-layer from $\mathcal{L}_0$ to $\mathcal{L}_{n-1}$, as explained in Section 2.3. The candidate compositions for layer $\mathcal{L}_n$ are generated from histograms of co-occurrences of $\mathcal{L}_{n-1}$ part activations within the time window τ_W (see also Equation (4)). Frequent co-occurrences indicate the presence of underlying patterns. New compositions are formed from combinations of $\mathcal{L}_{n-1}$ parts where the number of co-occurrences exceeds the learning threshold τ_C. The composition parameter μ is estimated from the corresponding histogram.

The $\mathcal{L}_1$ candidate compositions are thus constructed as a relative structure of two co-occurring $\mathcal{L}_0$ part activations, both occurring within the time window τ_W. This procedure is repeated on all consecutive layers, where activations of parts co-occurring within the time window on a previous layer $\mathcal{L}_{n-1}$ compose new part candidates on the next layer $\mathcal{L}_n$. Since the model allows for partial overlapping of the covered structure (e.g., P_1^2 in Figure 1), the structures on these layers represent 3–4 music events. Consequently, the $\mathcal{L}_N$ candidate compositions include all combinations of $\mathcal{L}_{N-1}$ part pairs representing structures of 2^{N-1}–2^N music events.

In the second step, a subset of compositions from $\mathcal{C}$ that covers a maximum number of events in the input data is selected. As the problem of selecting a set of compositions from $\mathcal{C}$ which optimally cover the input data is NP (nondeterministic polynomial time) complete, a greedy approach, which selects a subset of compositions and leaves a minimal amount of events in the input uncovered, was introduced in [41].

The composition selection uses part coverage as a measure of the part's suitability for selection. The coverage of the part P_i^n can be obtained by projecting its activations to the input layer and observing the covered events. For a single activation of the part P_i^n at the time T and the location L, coverage is defined as the union of coverages of its subparts:

$$C(A(P_i^n)) = \bigcup_{j=0}^{K-1} C(A(P_{k_j}^{n-1})). \tag{6}$$

When the input layer is reached, the coverage is defined by the presence of an event at the given location and time as:

$$C(A(P_1^0)) = \begin{cases} A_L(P_1^0): & A_M(P_1^0) > 0 \\ \emptyset: & otherwise \end{cases}. \tag{7}$$

Based on coverage, the greedy composition selection approach is defined as follows:

- the coverage of each part from $\mathcal{C}$ is calculated as a union of events in the training data covered by all activations of the part,
- parts are iteratively added to the new layer $\mathcal{L}_n$ by choosing the part that adds most to the coverage of the entire training set in each iteration. This ensures that only compositions that provide enough coverage of new data with regard to the currently selected set of parts will be added,
- the algorithm stops when the additional coverage falls below the learning threshold τ_L.

The learning procedure is repeated for each layer until a desired number of layers is reached. The reader should note that the number of layers governs the maximal length of encoded patterns, as discussed in the evaluation.

2.3. *Inferring Patterns*

A learned model captures the repetitive patterns in the training data, which are relatively encoded and may be observed through an inspection of the model's parts on its various layers. When a trained model is presented with new input data, the learned patterns may be located in the input through the process of inference. Inference calculates part activations on the input data (and thus absolute pattern positions) according to Equations (2) and (3). They are calculated bottom-up layer-by-layer, whereby the input data activates the layer $\mathcal{L}_0$. As already mentioned, the activation of a part represents a specific occurrence of the pattern it represents in the input. An activation has three components: location and onset time, which map the relative pattern onto a specific set of pitches within the input sequence of events (thus making it absolute), and magnitude, representing its strength. A part can concurrently activate at different locations, which indicates multiple occurrences of the represented pattern in the input representation.

Inference may be exact or approximate, where in the latter case two additional mechanisms, hallucination and inhibition, enable the model to find patterns with deletions, changes or insertions, thus increasing its predictive power and robustness.

2.3.1. Hallucination

As described in Section 2.1, a part activation is produced only if all subparts activate with magnitude greater than zero at locations which approximately correspond to the structure encoded by the part. This conservative behaviour may be relaxed by hallucination. It enables a part to produce activations even when the structure it represents is incomplete or modified in the input (e.g., missing notes, added notes, changed pitch, changed note order). Hallucination is important, as it enables the model to find variations of patterns represented by individual parts. The missing information is obtained from knowledge acquired during learning and encoded in the model structure. Using hallucination, the model generates activations of parts most fittingly covering the input representation, where notes which are not present, but are encoded in the model, are hallucinated. It is implemented by changing the conditions under which a part may activate. With hallucination, a part may activate even if all of its subparts are activated, when the percentage of events it represents, covered in the input, exceeds a hallucination threshold τ_H. Thus, if we set τ_H to one, the default behaviour is obtained, while lowering its value leads to increased hallucination and tolerance to changes in patterns.

The hallucination threshold τ_H influences the number of discovered patterns and identified pattern occurrences. When lowered, the amount of activations increases, as parts may activate on incomplete matches, thus producing activations which would otherwise not be generated. Additionally, if used during learning, the number of parts on lower layers will decrease, as parts added to a layer will have higher coverage due to more activations.

2.3.2. Inhibition

Inhibition in our model is a hypothesis refinement mechanism, which reduces the amount of redundant activations. An activation of a part P_i^n is inhibited (removed) when one or multiple parts $P_{j_1}^n, \ldots, P_{j_K}^n$ cover a large part of the same events in the input, but with stronger magnitude. More formally, activation of the part P_i^n is inhibited when the following conditions are met:

$$\exists \{P_{j_1}^n \ldots P_{j_K}^n\} : \frac{|C(A(P_i^n)) \setminus \bigcup_{k=1}^{K} C(A(P_{j_k}^n))|}{|C(A(P_i^n))|} < \tau_I \tag{8}$$

and

$$\forall P_{j_k}^n \in \{P_{j_1}^n \ldots P_{j_K}^n\} : A_M(P_{j_k}^n) > A_M(P_i^n). \tag{9}$$

The $C(A)$ represents activation coverage (Equation (6)), A_M activation magnitude (Equation (3)) and τ_I controls the strength of inhibition. If τ_I is set to zero, no inhibition occurs; the larger its value, the more activations are inhibited and propagated less between model layers. Notably, only activations with magnitude larger than that of the part P_i^n are considered in the inhibition process.

Besides reducing the number of activations and output patterns, the inhibition mechanism can also be used for producing alternative explanations of the input. If activations of the strongest pattern which inhibits other competing hypotheses are removed from the model, the next best hypothesis is selected during inference, thus providing an alternative explanation with different pattern occurrences to appear in the model's output.

3. Pattern Selection with SymCHM

The SymCHM model can be trained on a single or multiple symbolic music representations. It learns a hierarchical representation of patterns occurring in the input, where patterns encoded by parts on higher layers are compositions of patterns on lower layers. The inference produces part activations which expose the learned patterns (and their variations) in the input data. Shorter and more trivial patterns naturally occur more frequently, longer patterns less frequently. On the other hand, longer patterns may entirely subsume shorter patterns. Occurrences of melodic patterns in a given piece are discovered by observing activations of the learned model's parts, where each activation of a part is interpreted as an occurrence of the pattern encoded by the part.

To use the model for the discovery of repeated patterns and sections task, we need to select which of the found patterns will be provided in the model's output. In this Section, we present two approaches for a pattern selection.

3.1. Basic Selection

In a basic pattern selection, we output all patterns of sufficient complexity, as encoded by parts starting from the layer L up to the highest layer N. First, we select all parts from the layers $\mathcal{L}_L \ldots \mathcal{L}_N$. Since parts on higher layers are compositions of parts on lower layers, we exclude all parts which are subparts of a composition on a higher layer to avoid redundancy. The final selection of parts can be formulated as:

$$\bigcup_{l=L}^{N}\{P_i^l \in \mathcal{L}_l : (\neg\exists P_j^{l+1})[P_j^{l+1} \in \mathcal{L}_{l+1} \wedge P_i^l \in P_j^{l+1}]\} \tag{10}$$

Inference is then performed on a music piece and activations of the selected parts represent the found patterns and their locations in the piece. Hallucination and inhibition are applied during inference to provide balance between producing hypotheses which partially match the input representation (hallucination) and the amount of competitive hypotheses produced (inhibition).

3.2. SymCHMMerge: Improved Pattern Selection

An analysis of the basic pattern selection algorithm showed lack of diversity in the found patterns, as the patterns were often very similar and overlapping. We improved the algorithm by merging redundant patterns and adjusting the learning and inference parameters, and named the resulting model SymCHMMerge.

3.2.1. Merging Redundant Patterns

Since parts in our model are learned in an unsupervised manner, several parts may represent similar and overlapping patterns (e.g., patterns shifted by a few notes). Inhibition reduces redundant activations of such parts, however it is usually not enforced strongly, as it could overly reduce the number of activations and found patterns. To reduce the number of such overlapping patterns, we merge them into single, longer patterns.

Let $\pi(A(P_i^n))$ represent a pattern occurrence defined by the projection π of the activation A of the part P_i^n onto the layer $\mathcal{L}_0$. Ψ_i^n represents the set of all such pattern occurrences discovered by activations of the part:

$$\Psi_i^n = \bigcup_k \{\pi(A_k(P_i^n))\}. \tag{11}$$

Two pattern occurrences a_i and a_j, produced by the parts P_i^n and P_j^m, are taken to be redundant, if they overlap significantly. We express this by calculating the Jaccard similarity coefficient and compare it to a threshold τ_R:

$$\begin{gathered} a_i = \pi(A(P_i^n)), a_j = \pi(A(P_j^m)) \\ J(a_i, a_j) = \frac{|a_i \cap a_j|}{|a_i \cup a_j|} > \tau_R. \end{gathered} \tag{12}$$

We aim to merge redundant pattern occurrences of two parts if they frequently produce overlapping patterns. Therefore, we calculate the proportion of such patterns produced by the two parts as:

$$\frac{1}{|\Psi_i^n| + |\Psi_j^m|} \sum_{a_i \in \Psi_i^n} \sum_{a_j \in \Psi_j^m} |J(a_i, a_j) > \tau_R|. \tag{13}$$

If the proportion exceeds a threshold τ_M, all redundant pattern occurrences of the two parts are merged.

For evaluation, the thresholds τ_R and τ_M were both set to 0.5, meaning that pattern occurrences produced by two parts had to share at least 50% of events in the input layer and appear together in at least 50% of cases, to be merged.

3.2.2. Increasing Diversity

To address the problem of pattern diversity, we needed to increase the number of patterns found by the model. This was achieved with three simple adjustments. First, we lowered the candidate selection thresholds in the greedy phase of the learning process to add more parts to each layer (evaluation showed that on average 16% more parts were added). Second, more layers were considered when searching for pattern occurrences, and third, hallucination was increased during inference. All these modifications could also be made with the basic pattern selection approach; however, they would result in an even higher number of redundant patterns. With SymCHMMerge, redundant occurrences are merged and thus the diversity of the found patterns increases.

4. Evaluation

We evaluated the proposed model for the discovery of repeated themes and sections task in symbolic monophonic music pieces. Since we are searching for patterns within a given piece (and not across the entire corpus) the model was built independently for each piece and inferred on the same piece. All model parameters were kept constant during all evaluations and were not tuned to each specific case. The parameters were set to the values defined in Table 1. The τ_W parameter limiting the time span of activations was set to $\tau_W = 2^{n+2}$ events. The values and short descriptions of parameters are also listed in Table 1. The values for the τ_H and τ_I parameters are based on the stable performance achieved in the range around 0.5 for (see the Sensitivity to parameter values subsection. The τ_R and τ_M values were set to the majority thresholds of 50% and were not tuned. The τ_L parameter value was retained from the original spectral CHM where it was evaluated empirically.

Table 1. Model's parameter settings for the experiment.

Parameter	Description	Value
τ_H	Hallucination parameter retaining the activation of a part in an incomplete presence of the events in the input signal	0.5
τ_I	Inhibition parameter reducing the number of competing activations	0.4
τ_R	Redundancy parameter determining the the necessary amount of overlapping pattern occurrences in order for the occurrences to be merged	0.5
τ_M	Merging parameter determining the amount of redundant pattern occurrences needed for two patterns to be merged into one	0.5
τ_L	Learning threshold for added coverage which needs to be exceeded in order for a candidate composition to be retained while learning the model	0.005
τ_W	Window limiting the time span of activations, defined per layer $\mathcal{L}_n$	2^{n+2}

Table 2 shows the performance of SymCHM on the MIREX 2015 discovery of repeated themes and sections task. To compare SymCHM to SymCHMMerge, the Table 2 also includes the results of their evaluation on the publicly available JKU Patterns Development Dataset (PDD) [44]. Detailed results of SymCHMMerge on this dataset are shown in Table 3.

The JKU PDD dataset (the dataset is publicly available on this link: https://dl.dropbox.com/u/11997856/JKU/JKUPDD-Aug2013.zip) consists of five pieces:

- Bach's Prelude and Fugue in A minor (BWV(Bach-Werke-Verzeichnis) 889): 731 note events, 3 patterns, 21 pattern occurrences,
- Beethoven's Piano Sonata in F minor (Opus 2, No. 1), third movement: 638 note events, 7 patterns, 22 pattern occurrences,
- Chopin's Mazurka in B flat minor (Opus 24, No. 4): 747 note events, 4 patterns, 94 pattern occurrences,
- Gibbons' "The Silver Swan": 347 note events, 8 patterns, 33 pattern occurrences,
- Mozart's Piano Sonata in E flat major, K. 282-2nd movement: 923 note events, 9 patterns, 38 pattern occurrences.

Table 2. Evaluation of SymCHM, SymCHMMerge and Music Information Retrieval Evaluation eXchange (MIREX) results of other proposed approaches for the discovery of repeated themes and sections task on the JKU Patterns Development Dataset (PDD) and JKU Patterns Testing Dataset (PTD), denoted as MIREX 2015.

Algorithm	P_{est}	R_{est}	F_{1est}	$P_{occ(c=0.75)}$	$R_{occ(0.75)}$	$F_{1occ(c=0.75)}$	
SymCHM MIREX 2015	53.36	41.40	42.32	81.34	59.84	67.92	
NF1 MIREX 2014	50.06	54.42	50.22	59.72	32.88	40.86	
DM1 MIREX 2013	52.28	60.86	54.80	56.70	75.14	62.42	
OL1 MIREX 2015	61.66	56.10	49.76	87.90	75.98	80.66	
VM2 MIREX 2015	65.14	63.14	62.74	60.06	58.44	57.00	
SymCHM JKU PDD	67.92	45.36	51.01	93.90	82.72	86.85	
SymCHMMerge JKU PDD	67.96	50.67	56.97	88.61	75.66	80.02	
	TLF_1	$P_{occ(c=0.5)}$	$R_{occ(c=0.5)}$	$F_{1occ(c=0.5)}$	P	R	F_1
SymCHM MIREX 2015	37.78	73.34	62.48	67.24	10.64	6.50	5.12
NF1 MIREX 2014	33.28	54.98	33.40	40.80	1.54	5.00	2.36
DM1 MIREX 2013	43.28	47.20	74.46	56.94	2.66	4.50	3.24
OL1 MIREX 2015	42.72	78.78	71.08	74.50	16.0	23.74	12.36
VM2 MIREX 2015	42.20	46.14	60.98	51.52	6.20	6.50	6.2
SymCHM JKU PDD	51.75	78.53	72.99	75.41	25.00	13.89	17.18
SymCHMMerge JKU PDD	52.89	83.23	68.86	73.88	35.83	20.56	25.63

4.1. Evaluation Metrics

Evaluation metrics from the MIREX discovery of repeated themes and sections task were used for evaluation. This subsection provides a short description and formalization of the definitions found in the MIREX task definition [20]. The establishment measure (precision P_{est}, recall R_{est} and F score F_{1est}) evaluates the algorithm's ability to find at least one occurrence of each pattern shifted in time and pitch. Two occurrence measures F_{1occ} evaluate the extent of the model's ability to find all pattern occurrences, where the $c = \{0.5, 0.75\}$ factor represents the inexactness tolerance threshold. Meredith [30] proposed an additional three-layer metric (P_3, R_3, TLF_1) that provides balance between the establishment and the occurrence measures. The exact precision, recall and F score measures (P, R, F_1) show the algorithm's performance in matching the found patterns with the reference annotations in an exact manner.

The metrics are formally defined using the following set of symbols:

- $n_{\mathcal{P}}$: the number of patterns in a ground truth
- $\Pi = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_{n_{\mathcal{P}}}\}$: a set of ground truth patterns
- $\mathcal{P} = \{P_1, P_2, \ldots, P_{m_P}\}$—occurrences of pattern $\mathcal{P}$
- $n_{\mathcal{Q}}$: the number of patterns in the algorithm's output
- $\Xi = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_{n_{\mathcal{Q}}}\}$: a set of patterns returned by the algorithm
- $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_{m_{\mathcal{Q}}}\}$—occurrences of pattern $\mathcal{Q}$.
- k: the number of ground truth patterns identified by the algorithm

The standard precision is defined as $P = k/n_{\mathcal{Q}}$, the recall as $R = k/n_{\mathcal{P}}$, and the F_1 score as $F1 = 2 \times P \times R/(P + R)$. Due to the extreme difficulty of discovering strictly exact patterns, more robust versions of the metrics are provided: the occurrence and the establishment scores. First, the cardinality score is used to determine the music similarity between the annotated and the discovered patterns:

$$s_c(P_i, Q_j) : |P_i \cap Q_j| / \max\{|P_i|, |Q_j|\} \qquad (14)$$

A score matrix is calculated based on the similarity as follows:

$$s(\mathcal{P},\mathcal{Q}) = \begin{bmatrix} s(P_1,Q_1) & s(P_1,Q_2) & \dots & s(P_1,Q_{m_Q}) \\ s(P_2,Q_1) & s(P_2,Q_2) & \dots & s(P_2,Q_{m_Q}) \\ \vdots & \vdots & \ddots & \vdots \\ s(P_{m_P},Q_1) & s(P_{m_P},Q_2) & \dots & s(P_{m_P},Q_{m_Q}) \end{bmatrix} \tag{15}$$

Based on the score matrix, the establishment matrix is calculated from the set of annotated patterns Π and the set of algorithm's output patterns Ξ:

$$S(\Pi,\Xi) = \begin{bmatrix} S(\mathcal{P}_1,\mathcal{Q}_1) & S(\mathcal{P}_1,\mathcal{Q}_2) & \dots & S(\mathcal{P}_1,\mathcal{Q}_{n_Q}) \\ S(\mathcal{P}_2,\mathcal{Q}_1) & S(\mathcal{P}_2,\mathcal{Q}_2) & \dots & S(\mathcal{P}_2,\mathcal{Q}_{n_Q}) \\ \vdots & \vdots & \ddots & \vdots \\ S(\mathcal{P}_{n_\mathcal{P}},\mathcal{Q}_1) & S(\mathcal{P}_{n_\mathcal{P}},\mathcal{Q}_2) & \dots & S(\mathcal{P}_{n_\mathcal{P}},\mathcal{Q}_{n_Q}) \end{bmatrix} \tag{16}$$

The establishment precision is thus defined as:

$$P_{est} = \frac{1}{n_\mathcal{Q}} \sum_{j=1}^{n_\mathcal{Q}} \max\{S(\mathcal{P}_i,\mathcal{Q}_j) | i = 1 \dots n_\mathcal{P}\} \tag{17}$$

The establishment recall is defined as:

$$R_{est} = \frac{1}{n_\mathcal{P}} \sum_{j=1}^{n_\mathcal{P}} \max\{S(\mathcal{P}_i,\mathcal{Q}_j) | i = 1 \dots n_\mathcal{Q}\} \tag{18}$$

Additionally, the establishment F_1 score is calculated as:

$$F1_{est} = 2 \times P_{est} \times R_{est} / (P_{est} + R_{est}) \tag{19}$$

The establishment metrics reward a single match between the annotated and algorithm's patterns. To counterbalance this bias, the occurrence metrics are used. The occurrence metrics reward the algorithm's ability to find all occurrences of a single pattern. To loosen the exactness, the found patterns may be inexact. This inexactness is implemented using a threshold c (default values used in the 0.5 and 0.75), The indices $\mathcal{I}$ of the establishment matrix with values greater than or equal this threshold c are considered discovered. The occurrence matrix $O(\Pi,\Xi)$ is calculated using the following approach, starting with an empty $n_\mathcal{P} \times n_\mathcal{Q}$ matrix and the establishment indices $\mathcal{I}$:

$$\forall (i,j) \in \mathcal{I} : O(\Pi,\Xi)[i,j] = s(\mathcal{P}_i,\mathcal{Q}_j). \tag{20}$$

The occurrence precision score is consequently calculated using the occurrence matrix as follows:

$$P_{occ} = \frac{1}{n_{col}} \sum_{j=1}^{n_\mathcal{Q}} O(i,j) | i = 1 \dots n_\mathcal{P}, \tag{21}$$

where n_{col} represents the number of non-zero columns in occurrence matrix O. The occurrence recall score is analogously calculated as:

$$R_{occ} = \frac{1}{n_{row}} \sum_{j=1}^{n_\mathcal{P}} S(i,j) | i = 1 \dots n_\mathcal{Q}, \tag{22}$$

where n_{row} represents the number of non-zero rows in the occurrence matrix O.

4.2. Performance

The SymCHM with the basic pattern selection algorithm was submitted to the MIREX 2015 discovery of repeated themes and sections task. The results are shown in Table 2. The submitted model learned a six layer hierarchy, where activations of parts on Layers 4–6 were output as the found pattern occurrences.

In the MIREX 2015 evaluation [20], the two state-of-the art approaches by Velarde and Meredith (VM2) [32] and Lartillot (OL1) [34] achieved better overall results. However, the SymCHM outperformed other algorithms on the first piece in the MIREX evaluation dataset and achieved better results than VM2 in pattern occurrence measures, which indicated the model's ability to robustly identify the occurrences of the identified patterns. Compared to other approaches proposed in previous MIREX evaluations, such as NF1'14 [37] and DM1'13 [45], SymCHM found more pattern occurrences, as well a higher number of exact matches. The SymCHM also achieved a higher TLF_1 score when compared to NF1'14 submission.

Table 3. A detailed list of JKU Patterns Development Dataset results for the SymCHMMerge approach. The n_P and n_Q columns represent the number of annotated patterns and the number of discovered patterns respectively. Song names are shortened, using a four letter abbreviation of the composer's name.

Piece	n_P	n_Q	P_{est}	R_{est}	F_{1est}	$P_{occ(c=0.75)}$	$R_{occ(c=0.75)}$	$F_{1occ(c=0.75)}$
bach	3	2	100.00	66.67	80.00	100.00	45.65	62.68
beet	7	7	65.81	60.02	62.78	80.71	80.71	80.71
chop	4	5	47.95	49.81	48.86	62.36	51.96	56.69
gbns	8	3	78.16	35.49	48.81	100.00	100.00	100.00
mzrt	9	8	47.88	41.39	44.40	100.00	100.00	100.00
Average	6.2	5	67.96	50.67	56.97	88.61	75.66	80.02

Piece	P_3	R_3	TLF_1	$P_{occ(c=0.5)}$	$R_{occ(c=0.5)}$	$F_{1occ(c=0.5)}$	P	R	F_1
bach	62.96	41.97	50.37	100.00	45.65	62.68	100.00	66.67	80.00
beet	77.38	64.95	70.62	79.24	72.44	75.69	0.00	0.00	0.00
chop	46.96	39.92	43.15	57.00	46.29	51.09	0.00	0.00	0.00
gbns	81.82	34.33	48.37	100.00	100.00	100.00	66.67	25.00	36.36
mzrt	57.21	47.54	51.93	79.92	79.92	79.92	12.50	11.11	11.77
Average	65.27	45.74	52.89	83.23	68.86	73.88	35.83	20.56	25.63

To increase diversity and decrease redundancy, we introduced the SymCHMMerge with an improved pattern selection algorithm. Activations of parts on Layers 2–6 were considered for finding pattern occurrences, where each layer included 16% more parts on average due to the more relaxed learning conditions.

A comparison between both models on the JKU PDD dataset showed that the SymCHMMerge achieved significantly better results (Friedman's test: $\chi^2 = 7.2, p < 0.01$). It mostly improved in establishment measures, which indicated an improvement of the algorithm's ability to discover at least one occurrence of a pattern, tolerating for time shift and transposition [20]. On the other hand, occurrence measures $F_{1occ(c=0.75)}$ and $F_{1occ(c=0.5)}$ which evaluated the algorithm's ability to find all occurrences of the established patterns, have dropped by 5%. We attribute this drop to a higher number of established patterns, for which the occurrence measure is calculated. Finally, the absolute precision, recall and F scores significantly increased due to the SymCHMMerge's pattern merging procedure and increased pattern diversity.

4.3. Sensitivity to Parameter Values

To assess the sensitivity of SymCHMMerge to changes of model parameters, we analysed its performance by varying the inhibition and hallucination parameters τ_I and τ_H, which affect inference.

We observed the behaviour of occurrence and establishment measures in order to estimate the balance between the two. Due to the large number of possible parameter combinations, we evaluated how changes in one parameter (set for all layers) affect performance when all other parameters are fixed.

4.3.1. Inhibition

The top part of Figure 2 shows how changes in the inhibition parameter τ_I affect the results. An increase of τ_I increases inhibition and removes activations which are only partially covered by others, while a decrease will allow for more overlapping activations to propagate to higher layers. The plots show that reduced inhibition has a positive effect on occurrence recall, which is expected, as more activations are produced. It is even more interesting that it also positively affects precision of found occurrences, which might be explained by the fact that overlapping activations are successfully merged by the merging algorithm of SymCHMMerge. For the establishment metrics, the effect of changes in inhibition is not so obvious, and apart from extreme values, performance is stable.

4.3.2. Hallucination

The bottom part of Figure 2 shows how changes in the hallucination parameter τ_H affect performance. As described in Section 2.3.1, larger τ_H values decrease hallucination and thus the number of activations. Decreased hallucination affects both occurrence and establishment of patterns, as there is little tolerance for pattern variations. With more hallucination, both measures increase and then remain stable; again, precision is not affected significantly, as the merging algorithm of SymCHMMerge reduces the growing number of activations on higher layers.

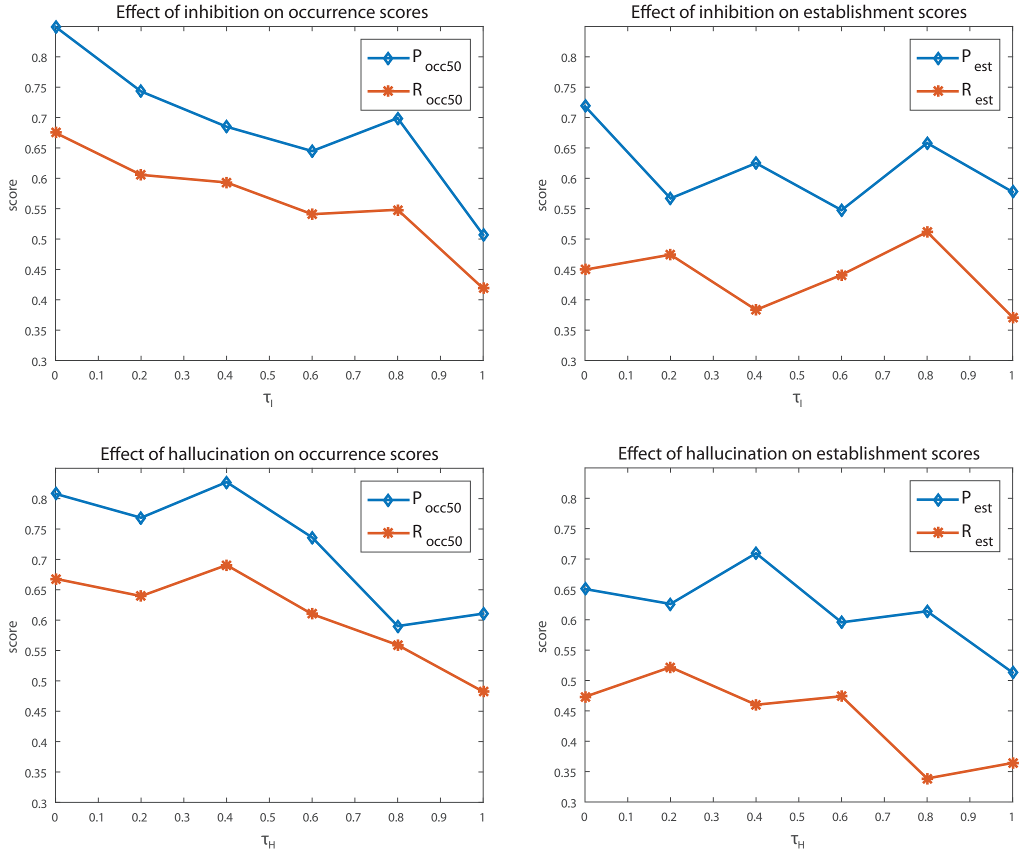

Figure 2. Sensitivity of the model to changes of the hallucination parameter τ_I (**top**) and the inhibition parameter τ_H (**bottom**). When one parameter was varied, all others remained fixed.

4.4. Error Analysis

To increase our understanding of the model's performance, we performed an analysis of its most common types of errors.

4.4.1. Incomplete Matches

We observed that the occurrence metrics increase when we allow for partially incomplete patterns to be discovered (hallucination), however, the exact F_1 scores do not always increase. After observing the pattern occurrences which do not contribute to the rise in F_1 score, we discovered that these patterns do not completely match the reference annotations, as shown in Figure 3.

The difference between a reference annotation and a model's proposed pattern usually presents itself at the edges of an occurrence, where the model assumes that one or more preceding or succeeding events belong to the pattern. These events frequently occur at the same locations (relative to the pattern), with similar time and pitch offsets. Thus, the model adds these events to the pattern occurrence, causing mismatch with the reference annotation. Such errors could be resolved by incorporating theoretical rules governing the beginnings and endings of patterns, e.g., gap rule ([46], p. 68) into the pattern selection algorithm.

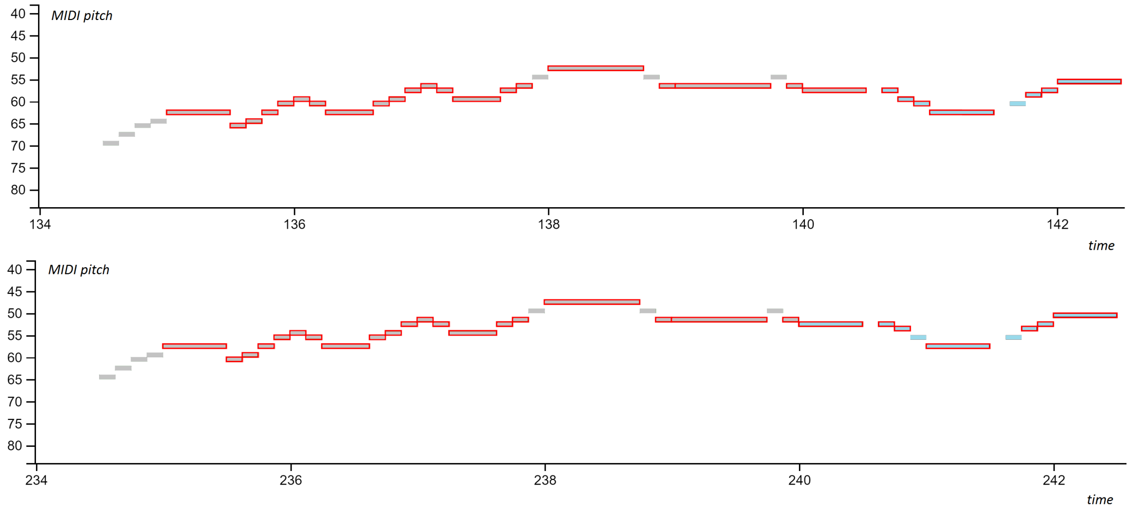

Figure 3. An incomplete pattern match of two pattern occurrences in Bach BWV 889 Fugue in A minor (from the JKU PDD dataset). Two pattern occurrences are presented in the figure (top and bottom). A piano roll representation is shown where the reference annotation is coloured in grey and the identified pattern occurrences outlined with red borders. Even though similar, events on the right side (shown in light blue) are not part of the reference annotations, however they are included in the model's patterns due to their co-occurrence with other events.

4.4.2. Unidentified Patterns

Patterns which were not identified by the model usually belong to one of two types: section patterns and short patterns.

Section patterns, such as in Mozart's Piano Sonata in E flat major, K. 282-2nd movement, remain unidentified. These section patterns represent large segments of music (50–137 events). The six layers in our model have the potential of encoding patterns of up to 64 events. While some of the reference patterns could be identified, the model did not contain a sufficient amount of layers to cover the largest patterns. We consequently focused on observing the absence of the shorter section patterns (between 50 and 64 events). While incomplete (often overlapping) matches of these patterns were found on the $\mathcal{L}_5$ and $\mathcal{L}_6$ layers (sub-patterns), there were no complete matches between the found patterns and the reference annotations. Furthermore, the overlap was not high enough that these sub-patterns would be merged during pattern merging.

The second subgroup—the short patterns—also frequently occur in evaluation datasets. These patterns are 4–5 events long. They are identified by the model on the layers $\mathcal{L}_2$ and $\mathcal{L}_3$, and also form compositions on higher layers. If such larger compositions are present, the pattern selection procedure excludes the short patterns from the final output.

The discovery of larger patterns could be improved by building additional compositional layers while learning the model, and by adjusting the merging rules for long patterns. To find more short patterns, we could add additional criteria that would counterbalance the promotion of longer patterns during pattern selection. For example, the event duration could be used when considering the importance of short events.

4.5. Drawbacks of the Evaluation

To establish the effectiveness of the proposed model in the symbolic domain, we evaluated the model for the pattern discovery task, where a comparison between the SymCHM and other approaches is based on the JKU PDD and JKU PTD datasets. To avoid diminishing the MIREX's position of being an evaluation exchange and not a benchmarking framework, we focused our evaluation on the two variants of the compositional model we developed, the SymCHM and SymCHMMerge, as shown in Table 2.

As thoroughly discussed by Meredith [30], this MIREX task possesses many drawbacks and thus might not be the optimal tool for an algorithm comparison. However, it is rather difficult to create an experiment which would provide a clearer evaluation of the algorithm's performance. First, a definition of a pattern is vague; there are several sources gathered in the JKU datasets. Some of the patterns in the ground truth represent themes, while others represent entire sections. Without any prior knowledge about the goal (length of pattern, perhaps a ratio between the length and the variation within the pattern occurrences), the metrics are logically leaning towards awarding the approach which finds most occurrences of the discovered pattern. It seems impossible to design an algorithm capable of finding a "pattern" when the definition of a pattern varies among the annotators. The three-layer F score proposed by Meredith is a step towards a metric which provides the balance between the establishment and the occurrence metrics otherwise provided. Second, the size of the dataset presents a limitation: the combined JKU PDD and JKU PTD datasets represent ten (classical) musical pieces in total. It is thus difficult to claim the datasets provide a representative sample of any kind of music or genre. However, we acknowledge the incredible effort put in the creation of the datasets and the tasks; we believe the size of the datasets is affected by the effort needed. Nevertheless, we believe the MIREX discovery of repeated themes and sections task is currently the best currently available approximation of a performance evaluation for the pattern discovery in music.

5. Conclusions

In the paper, we presented the compositional hierarchical model for pattern discovery in symbolic music representations. The model calculates a hierarchical representation of melodic patterns in a music corpus with a statistically-based learning algorithm. It can be viewed as a transparent deep architecture, combining the ability of unsupervised learning of multi-layer hierarchies with a transparent structure that enables insight into the learned concepts. The inference process with hallucination and inhibition mechanisms enables the search for pattern variations.

We evaluated the model in the MIREX evaluation campaign and its improved pattern selection algorithm on the JKU PDD dataset, where we show that we can obtain favourable results with the improved version of the model. We showed that the model can be used for finding patterns in symbolic music and that it can learn to extract patterns in an unsupervised manner without hard-coding the rules of music theory. We have also demonstrated the transfer of the model from classification tasks based on audio representations to pattern extraction in the symbolic domain. The results obtained by the model are not on par with the best two performing algorithms. Nevertheless, the proposed model performs better than several other proposed approaches. As discussed in Section 4.5, this evaluation contains many potential drawbacks, but it is currently the best approximation for pattern discovery evaluation. The definition of the 'pattern' itself is elusive and may contain many different explanations, varying from strictly music-theoretical, to mathematical formalization. The human perception of patterns in music itself is too difficult to explain and incorporate in a single formalized task. However, with the proposed model, we have demonstrated that a deep transparent architecture can tackle the pattern discovery by employing unsupervised learning and may thus better approximate how listeners recognize patterns than the rule-based systems. Due to its transparency, the model is not only applicable to tasks where a single output is provided, but can also be used for exploration and pattern discovery by an expert. The model produces multiple hypotheses on several layers, which can be used as reference points in a deeper semi-automatic music analysis. We believe this further strengthens the model's usefulness to the wider MIR community.

In our future work, we will focus on improving the model. We plan to include event duration into pattern selection and merging and adapt the model for polyphonic pattern discovery. We could also introduce pattern ranking, similar to [32], and add music theory rules, as discussed in Section 4.4. The model's output could further be optimized by supervised training of model parameters, especially the number of layers in the hierarchy and the layers in the model's output. However, a sufficiently large annotated dataset is needed for such an optimization, significantly larger than the datasets currently used to evaluate the pattern discovery task.

The proposed approach can also be applied to identify similar and inexact patterns across larger corpora. We plan on evaluating the model in an inter-opus pattern discovery task, aiding the current research in tune family identification and folk music analysis. To tackle classification tasks, the model can be observed as a feature generator; thus, its output can be employed as an input to tune family analysis, similarity comparison or composer identification.

Author Contributions: M.P., A.L. and M.M. conceived of and designed the experiments. M.P. performed the experiments. M.P. and M.M. analysed the data. M.P., A.L. and M.M. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CHM	Compositional Hierarchical Model
SymCHM	Compositional Hierarchical model for Symbolic music representations
SymCHMMerge	An extension of the SymCHM using a pattern merging technique

References

1. Lerdahl, F.; Jackendoff, R. *A Generative Theory of Tonal Music*; MIT Press: Cambridge, MA, USA, 1983.
2. Hamanaka, M.; Hirata, K.; Tojo, S. Implementing "A Generative Theory of Tonal Music". *J. New Music Res.* **2006**, *35*, 249–277.
3. Hirata, K.; Tojo, S.; Hamanaka, M. Techniques for Implementing the Generative Theory of Tonal Music. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), Vienna, Austria, 23–30 September 2007.
4. Marsden, A. Schenkerian Analysis by Computer: A Proof of Concept. *J. New Music Res.* **2010**, *39*, 269–289.
5. Todd, N. A Model of Expressive Timing in Tonal Music. *Music Percept. Interdiscip. J.* **1985**, *3*, 33–57.
6. Farbood, M. Working memory and the perception of hierarchical tonal structures. In Proceedings of the International Conference of Music Perception and Cognition, Seattle, WA, USA, 23–27 August 2010; pp. 219–222.
7. Balaguer-Ballester, E.; Clark, N.R.; Coath, M.; Krumbholz, K.; Denham, S.L. Understanding Pitch Perception as a Hierarchical Process with Top-Down Modulation. *PLoS Comput. Biol.* **2009**, *4*, 1–15.

8. McDermott, J.H.; Oxenham, A.J. Music perception, pitch and the auditory system. *Curr. Opin. Neurobiol.* **2008**, *18*, 452–463.
9. Humphrey, E.J.; Bello, J.P.; LeCun, Y. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR), Porto, Portugal, 8–12 October 2012.
10. Rigaud, F.; Radenen, M. Singing Voice Melody Transcription using Deep Neural Networks. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), New York, NY, USA, 7–11 August 2016; pp. 737–743.
11. Jeong, I.Y.; Lee, K. Learning Temporal Features Using a Deep Neural Network and its Application to Music Genre Classification. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), New York, NY, USA, 7–11 August 2016; pp. 434–440.
12. Schluter, J.; Bock, S. Musical Onset Detection with Convolutional Neural Networks. In Proceedings of the 6th International Workshop on Machine Learning and Music, held in Conjunction with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD 2013, Prague, Czech Republic, 23–27 September 2013.
13. Battenberg, E.; Wessel, D. Analyzing Drum Patterns using Conditional Deep Belief Networks. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), Porto, Portugal, 8–12 October 2012; pp. 37–42.
14. Deng, J.; Kwok, Y.K. A Hybrid Gaussian-Hmm-Deep-Learning Approach for Automatic Chord Estimation with Very Large Vocabulary. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), New York, NY, USA, 7–11 August 2016; pp. 812–818.
15. Campilho, A.; Kamel, M. (Eds.) *Image Analysis and Recognition*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Gemany, 2012; Volume 7324.
16. Coward, E.; Drabløs, F. Detecting periodic patterns in biological sequences. *Bioinformatics* **1998**, *14*, 498–507.
17. Margulis, E.H. *On Repeat: How Music Plays the Mind*; Oxford University Press: Oxford, UK, 2014; p. 224.
18. Downie, J.S. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoust. Sci. Technol.* **2008**, *29*, 247–255.
19. Meredith, D.; Lemstrom, K.; Wiggins, G.A. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *J. New Music Res.* **2002**, *31*, 321–345.
20. The MIREX Discovery of Repeated Themes & Sections Task. Available online: http://www.music-ir.org/mirex/wiki/2015:Discovery_of_Repeated_Themes_%26_Sections (accessed on 19 June 2015)
21. Collins, T.; Thurlow, J.; Laney, R.; Willis, A.; Garthwaite, P.H. A Comparative Evaluation of Algorithms for Discovering Translational Patterns in Baroque Keyboard Works. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), Utrecht, Netherlands, 9–13 August, 2010; pp. 3–8.
22. Wang, C.I.; Hsu, J.; Dubnov, S. Music Pattern Discovery with Variable Markov Oracle: A Unified Approach to Symbolic and Audio Representations. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), Malaga, Spain, 26–30 October 2015; pp. 176–182.
23. Cambouropoulos, E.; Crochemore, M.; Iliopoulos, C.S.; Mohamed, M.; Sagot, M.F. A Pattern Extraction Algorithm for Abstract Melodic Representations that Allow Partial Overlapping of Intervallic Categories. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), London, UK, 11–15 September 2005; pp. 167–174.
24. Conklin, D.; Bergeron, M. Feature Set Patterns in Music. *Comput. Music J.* **2008**, *32*, 60–70.
25. Conklin, D.; Anagnostopoulou, C. Representation and Discovery of Multiple Viewpoint Patterns. In Proceedings of the 2001 International Computer Music Conference, Havana, Cuba, 18–22 September 2001; pp. 479–485.
26. Conklin, D. Melodic analysis with segment classes. *Mach. Learn.* **2006**, *65*, 349–360.
27. Rolland, P.Y. Discovering Patterns in Musical Sequences. *J. New Music Res.* **1999**, *28*, 334–350.
28. Owens, T. *Charlie Parker: Techniques of Improvisation*; Number Let. 1 in Charlie Parker: Techniques of Improvisation; University of California: Los Angeles, CA, USA, 1974.
29. Cambouropoulos, E. Musical Parallelism and Melodic Segmentation. *Music Percept. Interdiscip. J.* **2006**, *23*, doi;10.1525/mp.2006.23.3.249.
30. Meredith, D. Music Analysis and Point-Set Compression. *J. New Music Res.* **2015**, *44*, 245–270.

31. Meredith, D. COSIATEC and SIATECCompress: Pattern Discovery by Geometric Compression. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), Curitiba, Brazil, 4–8 November 2013; pp. 1–6.
32. Velarde, G.; Meredith, D. Submission to MIREX Discovery of Repeated Themes and Sections. In Proceedings of the 10th Annual Music Information Retrieval eXchange (MIREX'14), Taipei, Taiwan, 27–31 October 2014; pp. 1–3.
33. Velarde, G.; Weyde, T.; Meredith, D. An approach to melodic segmentation and classification based on filtering with the Haar-wavelet. *J. New Music Res.* **2013**, *42*, 325–345.
34. Lartillot, O. Submission to MIREX Discovery of Repeated Themes and Sections. In Proceedings of the 10th Annual Music Information Retrieval eXchange (MIREX'14), Taipei, Taiwan, 27–31 October 2014; pp. 1–3.
35. Lartillot, O. In-depth motivic analysis based on multiparametric closed pattern and cyclic sequence mining. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), Taipei, Taiwan, 27–31 October 2014; pp. 361–366.
36. Ren, I.Y. Closed Patterns in Folk Music and Other Genres. In Proceedings of the 6th International Workshop on Folk Music Analysis, Dublin, Ireland, 15–17 June 2016, ; pp. 56–58.
37. Nieto, O.; Farbood, M. MIREX 2014 Entry: Music Segmentation Techniques And Greedy Path Finder Algorithm To Discover Musical Patterns. In Proceedings of the 10th Annual Music Information Retrieval eXchange (MIREX'14), Taipei, Taiwan, 27–31 October 2014; pp. 1–2.
38. Nieto, O.; Farbood, M.M. Identifying Polyphonic Patterns From Audio Recordings Using Music Segmentation Techniques. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), Taipei, Taiwan, 27–31 October 2014; pp. 411–416.
39. Reber, A.S. *Implicit Learning and Tacit Knowledge : An Essay on the Cognitive Unconscious*; Oxford University Press: Oxford, UK, 1993.
40. Pesek, M.; Leonardis, A.; Marolt, M. A compositional hierarchical model for music information retrieval. In Proceedings of the International Conference on Music Information Retrieval (ISMIR), Taipei, Taiwan, 27–31 October 2014; pp. 131–136.
41. Pesek, M.; Leonardis, A.; Marolt, M. Robust Real-Time Music Transcription with a Compositional Hierarchical Model. *PLoS ONE* **2017**, *12*, doi:10.1371/journal.pone.0169411.
42. Fidler, S.; Boben, M.; Leonardis, A. Learning Hierarchical Compositional Representations of Object Structure. In *Object Categorization: Computer and Human Vision Perspectives*; Cambridge University Press: Cambridge, UK, 2009; pp. 196–215.
43. Meredith, D. *Method of Computing the Pitch Names of Notes in MIDI-Like Music Representations*; US Patent US 20040216586 A1, 4 November 2004.
44. Collins, T. *JKU Patterns Development Database*; August 2013. Available online: https://dl.dropbox.com/u/11997856/JKU/JKUPDD-Aug2013.zip (accessed on 13 September 2017)
45. Meredith, D. COSIATEC and SIATECCompress: Pattern Discovery by Geometric Compression. In Proceedings of the 9th Annual Music Information Retrieval eXchange (MIREX'13), Curitiba, Brazil, 4–8 November 2013.
46. Temperley, D. *The Cognition of Basic Musical Structures*; MIT Press: Cambridge, MA. USA, 2001.